

MCMC methods for inference in a mathematical model of pulmonary circulation

L. Mihaela Paun^{1*} | M. Umar Qureshi^{2†} | Mitchel Colebank^{2†} | Nicholas A. Hill^{2*} | Mette S. Olufsen^{2†} | Mansoor A. Haider^{2†} | Dirk Husmeier^{2*}

¹School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8SQ, UK

²Department of Mathematics, NC State University, Raleigh, NC 27695, USA

Correspondence

L. Mihaela Paun, School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8SQ, UK
Email: l.paun.1@research.gla.ac.uk

Funding information

Engineering and Physical Sciences Research Council (EPSRC) of the UK, Grant/Award Number: EP/N014642/1; National Science Foundation, Grant/Award Number: NSF-DMS # 1615820

This study performs parameter inference in a partial differential equations system of pulmonary circulation. We use a fluid dynamics network model that takes selected parameter values and mimics the behaviour of the pulmonary haemodynamics under normal physiological and pathological conditions. This is of medical interest as it enables tracking the progression of pulmonary hypertension. We show how we make the fluids model tractable by reducing the parameter dimension from a 55D to a 5D problem. The Delayed Rejection Adaptive Metropolis (DRAM) algorithm, coupled with constraint nonlinear optimization is successfully used to learn the parameter values and quantify the uncertainty in the parameter estimates. To accommodate for different magnitudes of the parameter values, we introduce an improved parameter scaling technique in the DRAM algorithm. Formal convergence diagnostics are employed to check for convergence of the Markov chains. Additionally, we perform model selection using different information criteria, including Watanabe Akaike Information Criteria.

KEYWORDS

Pulmonary hypertension, Parameter Inference, Constraint Nonlinear Optimization, Delayed Rejection Adaptive Metropolis, Partial Differential Equations, Windkessel model

1 | INTRODUCTION

The cardiovascular circulation is composed of the systemic circulation and the pulmonary circulation. Extensive work has been done to model the systemic circulation (Shi et al., 2011), however pulmonary hypertension is one of the leading causes of right heart failure (Noordegraaf and Gali, 2011). Our work focuses on predicting the observed haemodynamic behaviour in the pulmonary circulation under normal and pathological conditions (hypoxia). Hypoxia is a pathological condition in which the body tissues are not sufficiently well oxygenated, leading to pulmonary hypertension.

Pulmonary hypertension (PH) is characterised by high mean blood pressure in the lungs (above 25 mmHg). PH leads to vascular remodelling, including stiffening, thickening, constriction of the small and large pulmonary arteries and microvascular rarefaction (Wang and Chesler, 2011). Microvascular rarefaction occurs in patients suffering of pulmonary hypertension, and is a pathological condition in which there are fewer capillaries per unit volume of body tissue. Developing a reliable predictive model for the pulmonary haemodynamics allows us to assist clinicians in diagnosing and treating pulmonary hypertension in a systematic manner. In addition, it helps reduce the number of invasive procedures for patients as, currently, pulmonary pressure is measured invasively via right heart catheterization (Tabima et al., 2012). In our problem we test the model in the context of data from mice, however, the work can be extended to human data.

One of the most important aspects in cardiovascular modelling is parameter inference and uncertainty quantification (Eck et al., 2016). This is because the number of parameters increases significantly when modelling several interconnected components at different scales. What is more, the limited amount of data available makes parameter estimation challenging in terms of finding physiologically sensible parameters.

Inferring key parameters for disease diagnosis and treatment planning is therefore an essential and challenging step in predicting the observed haemodynamics. One such parameter is the arterial stiffness, which is significantly higher for patients having pulmonary hypertension. Often, the parameters cannot be measured in-vivo, which creates the need for them to be learnt indirectly from the blood flow and pressure measurements.

In this paper, we present statistical techniques for inference of parameters of a mathematical model in order to reliably simulate and predict blood pressure and flow in the lungs. We first employ a nonlinear constraint optimization scheme to find the maximum likelihood estimates. We then quantify the uncertainty around these estimates by approximately sampling parameters from their posterior distribution using the Delayed Rejection Adaptive Metropolis (DRAM) algorithm and the Adaptive Metropolis (AM) algorithm.

Simulating blood flow and pressure in the 1D fluid-structure model described in the next section, requires solving a system of nonlinear partial differential equations (PDEs), which gets computationally expensive as the number of vessels increases in the model or a component model is made more complex. Therefore, starting the MCMC algorithm from optimised values allows us to save computational time. The goal of this paper is to show how optimization, combined with uncertainty quantification, can be used to learn the parameters relevant for pulmonary disease detection and, thus, to reliably predict the measured blood flow and pressure in the lungs. We show how by making a few restrictive assumptions about the parameters characterising the vessels' geometry or the boundary conditions in the PDEs, we manage to reduce the parameter dimension from a 55D problem to 5D. Furthermore, we illustrate via pseudocode how to deal with different parameter magnitudes by incorporating a parameter scaling technique in the MCMC Matlab toolbox (Laine, 2007). We test this code on real data coming from a healthy and a hypoxic mouse. Furthermore, we show the importance of parameter scaling on a concrete example, and the consequences of not including the scaling in the analysis.

The remainder of the paper is structured as follows: Section 2 introduces the mathematical model, Section 3 describes the statistical model used in this study, Section 4 provides an overview of the methodology used, including the

DRAM algorithm allowing for parameter scaling, Section 5 gives a few details of the computer simulations, Section 6 presents results of the methods applied on real data, and we finalise the paper by a discussion of the main findings of the study, as well as limitations and further work.

2 | MATHEMATICAL MODEL

This study predicts pulmonary arterial flow and pressure using a 1D fluid dynamics network model, described in detail in our recent studies Qureshi et al. (2017b) and Qureshi et al. (2017a). The model is derived from the incompressible axisymmetric Navier–Stokes equations for a Newtonian fluid, and coupled with a constitutive wall model predicting elasticity (i.e. stiffness) of the blood vessels. In addition, the model assumes that the vessels are cylindrical and tapering, and that the wavelength is significantly longer than vessels' radii. We use a tapering factor for the large arteries, as there is evidence of the vessels radii decreasing along their length and this was not quantified during the segmentation process. Under these assumptions, conservation of mass and momentum give

$$\frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} = 0, \quad \frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q^2}{A} \right) + \frac{A}{\rho} \frac{\partial p}{\partial x} = - \frac{2\pi\mu r}{\delta} \frac{q}{A}, \quad (1)$$

where x (cm) and t (s) are the axial and temporal coordinates, p (mmHg) is the blood pressure, q (ml/s) is the blood flow rate, A (cm²) is the cross-sectional area, $\rho = 1.055$ g/ml is the blood density, $\mu = 0.049$ g/(cm s) is the viscosity and $\delta = \sqrt{\mu T / 2\pi\rho}$ cm is the boundary-layer thickness of velocity profile.

The arterial walls are assumed to be homogeneous, isotropic and thin. We use a linear wall model, where the relationship between pressure, p (stress) and cross-sectional area A deformation (strain) can be expressed as follows:

$$p = p_0 + \frac{4}{3} f \left(1 - \sqrt{\frac{A_0}{A}} \right) \quad (2)$$

where A_0 is the unstressed vessel cross-sectional area, and f (mmHg) is the arterial network stiffness.

The flow in the network is driven by prescribing a flow waveform measured in the main pulmonary artery (MPA) of a healthy and hypoxic mouse. Conservation of blood flow ($q_p = q_{d_1} + q_{d_2}$) and continuity of pressure ($p_p = p_{d_1} = p_{d_2}$) are enforced at the bifurcations (p - parent, d_1 and d_2 - daughters), and an outflow condition predicting vascular bed impedance is prescribed using 3-element Windkessel model, with two resistors R_1 , R_2 and a capacitor C , of the form

$$Z(\omega) = R_1 + \frac{R_2}{1 + i\omega C R_2} \implies p(L, t) = \frac{1}{T} \int_0^T q(L, t - \tau) Z(\tau) d\tau, \quad (3)$$

The system of equations (1) and (2) were solved numerically using the two step Lax-Wendroff scheme (Lax and Wendroff, 1960). Simulations were set up to fit pressure waveforms measured in the MPA in a healthy and hypoxic mouse.

The coupled model contains a total of 100 parameters, used to determine pressure and flow in the network, associated with the vessel fluid dynamics (24 parameters), the geometry (43 parameters), and the outflow boundary conditions (33 parameters). Of these, 45 parameters describing physical properties are fixed. The remaining 55 parameters characterize vessel stiffness (21 parameters - one per vessel), outflow boundary conditions (33 parameters, 3 parameters per terminal vessel), and a tapering factor.

To reduce the dimension of the parameter space, we assume that the vessel stiffness f is constant throughout the network (Krenz and Dawson, 2003) (reducing the dimension by 20). Second, as described in Qureshi et al. (2017a), we

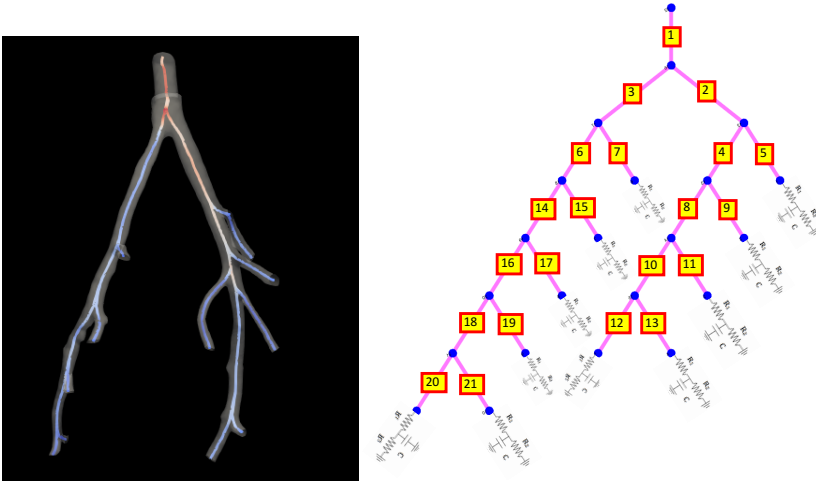


FIGURE 1 3D smoothed network from a micro-CT image of a healthy mouse lung (left) and the connectivity graph of the same network (right). A similar network was extracted for the hypoxic mouse (not shown here) with the same connectivity graph. Note, the connectivity graph only shows the network structure and does not indicate length and radii of individual vessels. A three-element Windkessel model with two resistors and a capacitor is attached at the end of each terminal vessel. Detailed network dimensions for healthy and hypoxic mouse can be found in Qureshi et al. (2017a).

predict the nominal outflow boundary condition for each vessel from estimates of total resistance and compliance and use a constant scaling factor to optimize each of the resistances and capacitances in the Windkessel models (reducing the dimension by 30). Briefly, total system resistance $R_T \equiv R_1 + R_2 = \bar{p}/\bar{q}$, where $R_1 = 0.2R_T$ and total capacitance C_T is estimated from diastolic pressure decay. These quantities are distributed to each terminal vessel, indexed j , as described in Qureshi et al. (2017a), giving a nominal set of parameters $\theta_{\text{wvk}}^j = \{R_{01}^j, R_{02}^j, C_0^j\}$. Introducing the scaling factors r_1 , r_2 and c , final Windkessel parameters are estimated by optimizing these scaling factors as

$$R_1^j = (1 - 0.5r_1)R_{01}^j, \quad R_2^j = (1 - 0.5r_2)R_{02}^j \quad \text{and} \quad C^j = (1 - 0.5c)C_0^j, \quad (4)$$

To account for the physiologically observed tapering in large arteries, a tapering factor ξ is introduced, such that

$$r_B = r_T(1 - 0.5\xi), \quad (5)$$

where r_T and r_B are the inlet and outlet radii of the vessels.

Above assumptions and criteria has allowed us to reduce the dimension of free parameters from 55 to 5. The parameters to be estimated include: vessel stiffness f , scaling factors r_1 , r_2 , c (equation (4)) and the tapering factor ξ .

Two network configurations are analysed:

$$\begin{aligned} \text{Network with straight vessels (4D model):} \quad & \theta = \{f, r_1, r_2, c\} \\ \text{Network with tapered vessels (5D model):} \quad & \theta = \{f, r_1, r_2, c, \xi\} \end{aligned} \quad (6)$$

3 | STATISTICAL MODEL

We start the analysis by specifying the statistical model:

$$\begin{aligned} y_i^q &= m_q(x_i; \theta) + \epsilon_{1i}, \\ y_i^p &= m_p(x_i; \theta) + \epsilon_{2i}, \end{aligned} \quad (7)$$

where

- $\mathbf{y}^q = [y_1^q, y_2^q, \dots, y_n^q]$ are the noisy measured flows, and $\mathbf{y}^p = [y_1^p, y_2^p, \dots, y_n^p]$ are the noisy measured pressures,
- $m(\cdot)$ describes the system behaviour that comes from numerically solving the fluids model equations (1) - (2), i.e. $m_q(\cdot)$ is the output (simulated) flow and $m_p(\cdot)$ is the output (simulated) pressure,
- θ are the parameters to be inferred from the observed flow and pressure. These relate to the Windkessel model parameters R_1, R_2, C defined in (3), to the vessel stiffness introduced in (2) and to the lung network geometry. For a more detailed description of the parameters see Section 2,
- $x_i \in \mathbf{x}$ denote other input variables (e.g. inflow into the MPA),
- ϵ are the errors, which we assume are i.i.d following a Gaussian distribution and are different for flow and pressure, due to different measurement techniques.

Our objective function is chosen to be the residual sum of squares (S), which measures the deviation of our simulated signal from the measured signal and which, under the Gaussian assumption of the errors, is the negative log likelihood of a Normal distribution. In this study, S is calculated for pressure in the main pulmonary artery, and we aim to find the parameters that minimise S. We can write our objective function for pressure as

$$S = (\mathbf{y}^p - m_p(\mathbf{x}; \theta))^2 = \sum_{i=1}^n (y_i^p - m_p(x_i; \theta))^2, \quad (8)$$

and the log likelihood of the pressure is

$$\log(L) = -n \log \sqrt{2\pi\sigma^2} - \frac{S}{2\sigma^2}. \quad (9)$$

For simplicity, in the remainder of the paper, we will no longer use the p index or the x_i term, but it should be noted that all the calculations are done for pressure, and \mathbf{x} is needed in solving the PDEs.

4 | STATISTICAL ANALYSIS METHODS

In this paper, we apply Markov Chain Monte Carlo (see Gilks et al. (1995) for a comprehensive review) to the problem of pulmonary circulation to approximately sample the parameters from the posterior distribution. Parameters are inferred using the Delayed Rejection Adaptive Metropolis (DRAM) algorithm (Haario et al., 2006) and we compare the performance of the DRAM algorithm with that of the Adaptive Metropolis (AM) algorithm (Haario et al., 2001). Since MCMC is too slow to converge when started from random points in parameter space, we first obtain point estimates using the maximum likelihood approach, and these estimates are used as initial values in the DRAM or AM algorithm. The concepts of optimization in a maximum likelihood context and uncertainty assessment of these estimates using

MCMC are reviewed in subsections 4.1 and subsubsection 4.2.1. Our work builds on and further improves on existing literature by allowing for a novel parameter scaling technique in the DRAM algorithm with informative priors for the parameters (see subsubsection 4.2.2). Following work in existing studies, we employ formal convergence diagnostics for the DRAM algorithm (Multivariate Potential Scale Reduction Factor, Geweke test, Integrated Autocorrelation Time and Effective Sample Size), as described in 4.3. The last step is to use existing scores for selecting the better model for explaining the data (AIC, AICc, BIC, DIC, WAIC), reviewed in subsection 4.4.

4.1 | Maximum likelihood with nonlinear optimization

We first apply nonlinear constraint optimization to find the global or local optimum of the objective function, as this exhibits unimodality in low dimensions (see Results section) and additionally, it is nonlinear in the parameters, which are bounded (see the Simulations section for a justification of the parameter ranges). We apply the Sequential Quadratic Programming (SQP) algorithm, which iteratively generates estimates that converge to a solution of (10) by constructing and solving quadratic sub-problems that approximate the nonlinear programming problem defined by (10).

$$\begin{aligned}
 &\text{minimize} && e(\theta), \theta \in \mathbb{R}^d, \text{ where } e(\theta) \text{ is the residual-sum-of-squares, } S \text{ (see (8))} \\
 &\text{subject to} && g(\theta) = 0, g : \mathbb{R}^d \rightarrow \mathbb{R}^t \text{ equality constraints defining relations} \\
 & && \text{between parameters,} \\
 & && h(\theta) \leq 0, h : \mathbb{R}^d \rightarrow \mathbb{R}^u \text{ inequality constraints defining a} \\
 & && \text{restricted parameter ranges.}
 \end{aligned} \tag{10}$$

The gradient-based method used here involves Taylor series expansion and, so, it requires the function to be twice continuously differentiable in the neighbourhood of the point tested and the Hessian matrix of e to be positive definite. This method has been shown to be a very good tool for solving highly nonlinear, fairly low dimensional problems, where we have equality and inequality constraints and where the function displays unimodality. In this algorithm, information about the function to be minimised, e , and about the equality and inequality functions g and h is stored in a Lagrangian function: $L(\theta, \Lambda, \Gamma) = e(\theta) + \sum_d \Lambda_d g_d(\theta) + \sum_d \Gamma_d h_d(\theta)$, where Λ and Γ are the Lagrange multipliers. Next, the Lagrangian function is minimised using Newton's method, due to the nonlinearity of the problem, which calls for numerically iterative methods (finding the critical points that make $\nabla L = 0$ is challenging since some of the derivatives may be highly nonlinear). We can optimise the Lagrangian function instead of the objective because the critical points of the Lagrangian are the same as those of the objective function. The problem is then turned into a quadratic subproblem. At iteration k , the direction, v , in which the algorithm searches the minimum is a solution to the equation:

$$\begin{aligned}
 &\min_v e(\theta_k) + \nabla e(\theta_k)^T v + \frac{1}{2} v^T \nabla_{\theta\theta}^2 L(\theta_k, \Lambda_k, \Gamma_k) v, \\
 &\text{s.t. } g(\theta_k) + \nabla g(\theta_k)^T v \geq 0 \text{ and } h(\theta_k) + \nabla h(\theta_k)^T v = 0.
 \end{aligned}$$

For more details of this algorithm, see Wilson (1967) and Boggs and Tolle (2000).

4.2 | Uncertainty quantification with MCMC

4.2.1 | Overview of MCMC (DRAM)

The Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953) is one of the most widely used MCMC methods. In this algorithm, new samples are generated from the proposal distribution and are either accepted or rejected depending on how likely it is that these samples come from the target distribution, i.e. accept with probability:

$$\alpha_1(\theta^{k-1}, \theta^*) = \min \left(1, \frac{\pi(\theta^*) q_1(\theta^*, \theta^{k-1})}{\pi(\theta^{k-1}) q_1(\theta^{k-1}, \theta^*)} \right), \quad (11)$$

where θ_{k-1} is the current point, θ^* the proposed point, $q_1(\theta^*, \theta^{k-1})$ the proposal distribution from θ^{k-1} to θ^* and $\pi(\cdot)$ the stationary distribution. If a symmetric proposal distribution q_1 (i.e. $q_1(\theta^*, \theta^{k-1}) = MVN(\theta^{k-1}, V_k)$), is used, then the ratio of the proposal distributions will be 1. A poor proposal distribution will result in a high rejection rate. To overcome this, upon rejection of θ^* , instead of retaining θ^{k-1} , one can make a second attempt to move (Tierney and Mira, 1999); in the 2^{nd} proposal, a different distribution is used (i.e. $q_2(\theta^{*2}, \theta^*, \theta^{k-1}) = MVN(\theta^{k-1}, \beta^2 V_k)$ for a symmetric proposal distribution) and this is allowed to depend on the rejected sample, which can be proved to not destroy the Markovian property of the chain (Green and Mira, 2001). Having started in θ^{k-1} and after rejecting θ^* , the second proposed point, θ^{*2} is accepted with probability:

$$\alpha_2(\theta^{k-1}, \theta^*, \theta^{*2}) = \min \left(1, \frac{\pi(\theta^{*2}) q_1(\theta^{*2}, \theta^*) q_2(\theta^{*2}, \theta^*, \theta^{k-1}) [1 - \alpha_1(\theta^{*2}, \theta^*)]}{\pi(\theta^{k-1}) q_1(\theta^{k-1}, \theta^*) q_2(\theta^{k-1}, \theta^*, \theta^{*2}) [1 - \alpha_1(\theta^{k-1}, \theta^*)]} \right), \quad (12)$$

The acceptance probability in the 2^{nd} stage is calculated in such a way that detailed balance is preserved. The ratio of the q_2 proposal distributions will become 1 if the proposal distribution is symmetric. The Delayed Rejection (DR) algorithm may have a smaller rejection rate, which implies a smaller asymptotic variance of the estimates compared to the MH algorithm (Mira, 2001a).

In cases where the parameters to be learnt are strongly correlated, the parameter dimensionality is large or the problem is highly nonlinear in the parameters, then running the classical MH algorithm can be inefficient, since the chain will move slowly in the parameter space due to the random walk behaviour of the algorithm, thus taking a long time to approach the target density. A solution to this is to adapt the Gaussian proposal covariance depending on the shape (or spatial orientation) and size of the target distribution, i.e. we adapt the proposal covariance matrix, V_k based on the past chain (Haario et al., 2001). It can be shown that this Adaptive Metropolis (AM) algorithm still produces an ergodic Markov chain (Haario et al., 2001).

The proposal distribution in the AM algorithm is a Gaussian distribution centred at the current point θ^{k-1} and covariance matrix equal to $V_k = V_k(\theta^0, \dots, \theta^{k-1})$. The proposal covariance matrix, $V_k = R_k^T R_k$ is adapted at given intervals, after possibly some non-adaptation time t_{ad} , according to the following formula:

$$V_k = \begin{cases} V_0, & \text{if } k \leq t_{ad} \\ s_d \text{cov}(\theta^0, \dots, \theta^{k-1}) + \epsilon I_d, & \text{if } k > t_{ad}, \end{cases} \quad (13)$$

where V_0 is the initial proposal covariance matrix, s_d is a parameter that depends on the dimension d of the target ($s_d = 2.4^2/d$ has been shown to be optimal for Gaussian targets in terms of mixing) (Haario et al., 2001), $\epsilon > 0$ is a constant that ensures that V_k does not become singular, and $(\theta^0, \dots, \theta^{k-1})$ are the past chain samples.

DRAM (Haario et al., 2006) increases the convergence through the use of the rejected values in the DR step (local

adaptation), and of the past chain samples to learn about the posterior distribution through the adapted covariance matrix, in the AM step (global adaptation).

4.2.2 | Delayed Rejection Adaptive Metropolis (DRAM) with parameter scaling

A pseudocode of the DRAM algorithm can be found in Smith (2013) (pages 175–176). In practice, we implement the algorithm by making use of the Matlab DRAM toolbox (Laine, 2007). However, the toolbox does not account for the case where the parameters differ by orders of magnitude, and so the code had to be modified to allow scaling the parameters. The pseudocode for DRAM with an informative prior, together with the parameter scaling adjustment can be found in Algorithms 1–4. Here we improve the Algorithms 8.8 and 8.10 in Smith (2013). We introduce a novel extension to DRAM by adding an informative prior and improving the scaling of the parameters. This improvement concerns the way that the acceptance probabilities are calculated. More precisely, the jump densities should be calculated for the transformed parameters (see step 12 in Algorithm 4). Using the original parameters can cause the ratio of the jump densities to become very large in exceptional cases, making us falsely accept the proposals. For example, if the starting value, which is also the optimum, is close to the boundaries, the algorithm will often jump outside boundaries. This will make some parameters get very high values (e.g. in the second DR try), and so the density in the numerator will become much higher than the density in the denominator in the jump ratio (see equation (12)). This, however, does not represent a problem for the first stage acceptance probabilities when a symmetric proposal distribution is used, as the proposals will cancel out in the ratio. Hence, the correction only affects DRAM and DR, while no correction is needed for AM.

Algorithm 1 Delayed Rejection Adaptive Metropolis Algorithm (DRAM) with informative prior (no hyperparameters)

- 1: Initialise design parameters: M : number of chain iterations, β : scaling factor for proposal covariance matrix in DR, n_{dr} : number of DR tries, t_{ad} : adaptation interval, b_s : burn-in scale, b_t : burn-in time, γ_s^2 : prior for σ^2 (error variance), n_s : prior accuracy for γ_s^2 .
- 2: Compute $\theta^0 = \arg \min_{\theta} \sum_{j=1}^n [y_j - m_j(\theta)]^2$, where $a_j \leq \theta_j \leq b_j, j = 1 \dots d, d$: number of parameters, using nonlinear constraint optimization (SQP algorithm) with 20 overdispersed starting values generated from a Sobol sequence.
- 3: Initialise $\sigma_0^2 = \frac{S_0}{n-d}$, where S_0 : optimised residual-of-squares value, n : number of observations.
- 4: Assume that the prior distribution for θ is a Multivariate Normal distribution (independent θ_s), $\theta \sim \mathcal{MVN}(\mu, T)$
- 5: Set initial $S_{\text{prior}}^0 = \sum_{j=1}^d \frac{(\theta_j^0 - \mu_j)^2}{t_j^2}$.
- 6: Either set the initial covariance matrix V_0 to be the Hessian matrix from optimization or set V_0 via monitoring of acceptance rate.
- 7: Set $R_0 = \text{chol}(V_0)$, where "chol" stands for the Choleski factorization, as an efficient numerical solution of $V_0 = R_0^T R_0$.
- 8: **for** $k = 1, \dots, M$ **do**
- 9: Sample $u_k \sim \mathcal{MVN}(0, I)$.
- 10: Construct candidate from Gaussian proposal: $\theta^* = \theta^{k-1} + R_{k-1} u_k$.
- 11: **if** $a_j \leq \theta_j^* \leq b_j$, for all j **then** ▷ outside boundaries
- 12: Jump to DR Algorithm 3 for iteration k
- 13: **else**
- 14: Calculate $S_{\theta^*} = \sum_{j=1}^n [y_j - m_j(\theta^*)]^2$, $S_{\text{pri}}^* = \sum_{j=1}^d \frac{(\theta_j^* - \mu_j)^2}{t_j^2}$
- 15: **end if**
- 16: Calculate the acceptance probability,

$$\alpha_1(\theta^*, \theta^{k-1}) = \min \left(1, e^{\left[-0.5 \left(\frac{S_{\theta^*} - S_{\theta^{k-1}}}{\sigma_{k-1}^2} + S_{\text{pri}}^* - S_{\text{pri}}^{k-1} \right) \right]} \right).$$
- 17: Sample $v_{\alpha_1} \sim \mathcal{N}(0, 1)$.
- 18: **if** $v_{\alpha_1} < \alpha_1$ **then**
- 19: $\theta^k = \theta^*, S_{\theta^k} = S_{\theta^*}, S_{\text{pri}}^k = S_{\text{pri}}^*$
- 20: **else**
- 21: Enter DR Algorithm 3 for iteration k
- 22: **end if**
- 23: Update $\sigma_k^2 \sim \text{Inv-Gamma} \left(\frac{n_s + n}{2}, \frac{n_s \gamma_s^2 + S_{\theta^k}}{2} \right)$
- 24: Enter AM Algorithm 2 for iteration k
- 25: **end for**

Algorithm 2 Adaptive Metropolis Algorithm (AM)

```

1: if  $k < b_t$  then                                     ▶ during burn-in time, no adaptation, just scaling R
2:    $R_k \leftarrow R_{k-1} / b_s,$                              ▶ decrease R if rejection rate > 0.95
3:    $R_k \leftarrow b_s R_{k-1},$                              ▶ increase R if rejection rate < 0.05
4: else
5:   if  $\text{mod}(k, t_{\text{ad}}) = 1$  then
6:     Set  $V_k = s_d \text{cov}(\theta^0, \theta^1, \dots, \theta^k) + \epsilon I_d, R_k = \text{chol}(V_k)$ 
7:   else
8:      $V_k = V_{k-1}, R_k = R_{k-1}$ 
9:   end if
10: end if
11: Output  $R_k$ 

```

Algorithm 3 Delayed Rejection Algorithm (DR) for $n_{\text{dr}} = 2$, i.e. for 2^{nd} stage proposals

```

1: Sample  $\mathbf{u}_k \sim \mathcal{MVN}(0, I)$ .
2: Construct  $2^{nd}$  stage candidate from Gaussian proposal:  $\theta^{*2} = \theta^{k-1} + \beta R_{k-1} \mathbf{u}_k$ .
3: if  $a_j \leq \theta_j \leq b_j$ , for all  $j$  then
4:    $\theta^k = \theta^{k-1}, S_{\theta^k} = S_{\theta^{k-1}}, S_{\text{pri}}^k = S_{\text{pri}}^{k-1}$ 
5:   and jump to AM Algorithm 2.
6: else
7:    $S_{\theta^{*2}} = \sum_{i=1}^n [y_i - m_i(\theta^{*2})]^2, S_{\text{pri}}^{*2} = \sum_{j=1}^d \frac{(\theta_j^{*2} - \mu_j)^2}{t_j^2}$ 
8: end if
9: Calculate the acceptance probability,  $\alpha_2(\theta^{*2}, \theta^*, \theta^{k-1})$  according to equation (12).
10: Sample  $v_{\alpha_2} \sim \mathcal{N}(0, 1)$ .
11: if  $v_{\alpha_2} < \alpha_2$  then,
12:    $\theta^k = \theta^{*2}, S_{\theta^k} = S_{\theta^{*2}}, S_{\text{pri}}^k = S_{\text{pri}}^{*2}$ 
13: else
14:    $\theta^k = \theta^{k-1}, S_{\theta^k} = S_{\theta^{k-1}}, S_{\text{pri}}^k = S_{\text{pri}}^{k-1}$ 
15: end if

```

Algorithm 4 Delayed Rejection Adaptive Metropolis (DRAM) with scaled parameters

- 1: Denote $\theta_s = \theta./s$, where s are the scaling factors.
- 2: In Algorithm 1:
- 3: DRAM 2: Compute $\theta^0 = \arg \min_{\theta} \sum_{i=1}^n [y_i - m_i(\theta \times s)]^2$
- 4: DRAM 10: Construct candidate from Gaussian proposal:

$$\theta_s^* = \theta_s^{k-1} + R_{k-1} u_k.$$

and set

$$\theta^* = \theta_s^* \times s$$

- 5: DRAM 14: Set

$$S_{\text{pri}}^* = \sum_{j=1}^d \frac{(\theta_j^* - \mu_j)^2}{t_j^2} + \log(\det(J))$$

where J is the Jacobian of the parameter transformation. If the parameters are assumed independent and s is constant, then $\log(\det(J)) = \sum_{j=1}^d s_j$.

- 6: DRAM 16: Calculate the acceptance probability,

$$\alpha_1(\theta_s^*, \theta_s^{k-1})$$

- 7: **Note:** if a noninformative prior is used and s is a constant, then we are left with the likelihood contribution in the acceptance probability, i.e. $\alpha = \min \left(1, e^{-0.5 \left[\frac{S_{\theta^*} - S_{\theta^{k-1}}}{\sigma_{k-1}^2} \right]} \right)$. In addition, in the likelihood calculations, we always use the unscaled parameters, whereas the priors and proposals are done in the transformed parameter space.

- 8: DRAM 19: In addition, set $\theta_s^k = \theta_s^*$.

- 9: In Algorithm 3:

- 10: DR 2: Construct 2^{nd} stage candidate from Gaussian proposal:

$$\theta_s^{*2} = \theta_s^{k-1} + \beta R_{k-1} u_k,$$

and set

$$\theta^{*2} = \theta_s^{*2} \times s$$

- 11: DR 7: Set

$$S_{\text{pri}}^{*2} = \sum_{j=1}^d \frac{(\theta_j^{*2} - \mu_j)^2}{t_j^2} + \log(\det(J))$$

- 12: DR 9: Calculate the acceptance probability in the scaled parameter space, $\alpha_2(\theta_s^{*2}, \theta_s^*, \theta_s^{k-1})$.

- 13: DR 12 or DR 14: In addition, set $\theta_s^k = \theta_s^{*2}$ or $\theta_s^k = \theta_s^{k-1}$.

Note: If a noninformative prior is used, then $S_{\text{pri}} = 0$ in Algorithms 1, 3 and 4.

4.3 | Convergence Diagnostics

Once the Markov chains are produced using the algorithms discussed above, we test for convergence, i.e. test whether our Markov chains have converged in distribution to the posterior distribution of interest.

4.3.1 | Integrated Autocorrelation Time (IACT)

One way to assess convergence is to assess the autocorrelations between the draws of the Markov chain. The lag l autocorrelation ρ_l is the correlation between every draw and its l^{th} lag:

$$\rho_l = \frac{\sum_{k=1}^{M-l} (\theta_k - \bar{\theta})(\theta_{k+l} - \bar{\theta})}{\sum_{k=1}^M (\theta_k - \bar{\theta})^2} \quad (14)$$

The Integrated Autocorrelation Time (Christen and Fox, 2010) is given by: $\tau = 1 + 2 \sum_{l=1}^{\infty} \rho_l$.

It would be expected that the l^{th} lag autocorrelation is smaller as l increases (2nd and 50th draws should be less correlated than 2nd and 4th draws). Relatively high autocorrelation for higher values of l suggests high degree of correlation between draws and thus, slow mixing.

Having IACT, we can also derive the Effective Sample Size (ESS) (Kass et al., 1998), which gives an estimate of the equivalent number of independent iterations that the chain represents. Low ESS can indicate that the sampler used is inefficient.

The formula for ESS is given by:

$$ESS = \frac{M}{1 + 2 \sum_{l=1}^{\infty} \rho_l} = \frac{M}{\tau}. \quad (15)$$

4.3.2 | Geweke test

The Geweke test (Geweke, 1992) formally tests for equality of the mean of two sub-chains (typically the first 10% and last 50% of the iterations). A low p-value for the Z-statistic will indicate non-convergence.

4.3.3 | Brooks Gelman Rubin test

The Brooks Multivariate Potential Scale Reduction Factor (MPSRF) (Brooks and Gelman, 1998) can be calculated for multiple chains from starting values overdispersed with respect to the stationary distribution. MPSRF is a multivariate extension to the Gelman-Rubin test (Gelman and Rubin, 1992), assessing convergence of the parameters simultaneously. Not only does this take into account the variances of individual parameters, but also the interactions between the parameters (the covariances). The MPSRF is calculated in such a way that it is an approximate estimate of the maximum upper bound of the univariate SRF for every parameter. Suppose we have m chains in parallel, and θ_j^k is the set of parameters drawn at the k^{th} iteration, $k = 1, \dots, M$ from the j^{th} , $j = 1, \dots, m$ chain. Then we can calculate the posterior variance-covariance matrix, as follows:

$$\hat{V} = \frac{M-1}{M} W + \left(1 + \frac{1}{m}\right) \frac{B}{M},$$

where

$$W = \frac{1}{m(M-1)} \sum_{i=1}^m \sum_{k=1}^M (\theta_i^k - \bar{\theta}_{i.})(\theta_i^k - \bar{\theta}_{i.})'$$

and

$$B = \frac{M}{m-1} \sum_{i=1}^m (\bar{\theta}_{i.} - \bar{\theta}_{..})(\bar{\theta}_{i.} - \bar{\theta}_{..})'$$

W is the within-chain covariance matrix and B the between-chain covariance matrix. If λ_1 is the largest eigenvalue of the symmetric and positive definite matrix $W^{-1}B$ (see proof in (Brooks and Gelman, 1998)), then the MPSRF,

$$\hat{R}^p = \frac{M-1}{M} + \frac{m+1}{m} \lambda_1$$

approaches 1 if $M \rightarrow \infty$ and the chains have converged to the stationary distribution, i.e. $W \approx B$ and, hence $\lambda_1 \rightarrow 0$. In addition, a MPSRF plot is useful in indicating whether the lack of convergence is due to the correlation between parameters.

4.4 | Model selection

Like in any other Bayesian application, where we have two or more competing models, we may be interested in comparing them, in order to choose the better model. A more complex model is not necessarily favoured, despite the fact that it provides a better fit. Penalising for complexity (too many parameters) is a necessary adjustment to prevent from choosing an overparameterised model that overfits the data and is not generalisable to future data. In this paper, we use the corrected Akaike Information Criterion (AICc) (Burnham and Anderson, 2002), the Bayesian Information Criteria (BIC) (Schwarz, 1978), the Deviance Information Criteria (DIC) (Spiegelhalter et al., 2002) and the Watanabe AIC (WAIC) (Watanabe, 2010) score for model selection, and the model with the lowest score will be preferred.

4.4.1 | AICc and BIC

AICc (Burnham and Anderson, 2002) and BIC (Schwarz, 1978) can be used for model selection in the maximum likelihood framework, where we obtain a maximum likelihood estimate. We use the maximum log likelihood (i.e. the minimum residual sum-of-squares, S) to represent the goodness of fit, and we apply a penalty to avoid overfitting. AICc and BIC penalise for the model complexity differently: AICc applies a lower penalty, while BIC tends to penalise more and choose simpler models. AICc is a modification of AIC (Akaike, 1971) by adding a correction for finite sample sizes, hence applying a higher penalty to avoid overfitting. The formula for AICc is given for a univariate Gaussian errors model:

$$\begin{aligned} AICc &= -2 \log(\mathbf{y}|\hat{\theta}_{\text{mle}}) + 2d + \frac{2d(d+1)}{n-d-1}, \\ BIC &= -2 \log(\mathbf{y}|\hat{\theta}_{\text{mle}}) + d \log n, \end{aligned} \tag{16}$$

where $\log(\mathbf{y}|\hat{\theta}_{\text{mle}})$ is the maximum log-likelihood, d is the number of parameters in the model and n is the total number of observations. We can derive these formulas to depend on the residual sum-of-squares and the formula involving S is

given in equation (17):

$$\begin{aligned} AICc &= \frac{S}{\sigma^2} + n \log \sigma^2 + 2d + \frac{2d(d+1)}{n-d-1} + C, \\ BIC &= \frac{S}{\sigma^2} + n \log \sigma^2 + d \log n + C, \end{aligned} \quad (17)$$

where $C = n \log 2\pi$ is a constant.

4.4.2 | DIC

DIC (Spiegelhalter et al., 2002) is a partially Bayesian version of AIC, where $\hat{\theta}_{\text{mle}}$ is replaced by the posterior mean $\hat{\theta}_{\text{Bayes}} = E(\theta|\mathbf{y})$ and the penalisation is data driven. Then,

$$DIC = -2 \log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) + 2p_{\text{DIC}}, \quad (18)$$

where

$$p_{\text{DIC}} = 2 \left(\log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) - E_{\theta|\mathbf{y}}(\log p(\mathbf{y}|\theta)) \right), \quad (19)$$

where $E_{\theta|\mathbf{y}}(\log p(\mathbf{y}|\theta))$ is the expectation of the log posterior predictive density, which in practice is calculated by replacing the expectation with the average over the posterior draws, as follows (Gelman et al., 2014):

$$\text{computed } p_{\text{DIC}} = 2 \left(\log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) - \frac{1}{S} \sum_{s=1}^S \log p(\mathbf{y}|\theta^s) \right), \quad (20)$$

where the sum is over the set of parameters θ^s , $s = 1, \dots, S$, that have approximately been drawn from the posterior distribution with MCMC (i.e. θ^s is the MCMC sample).

4.4.3 | WAIC

WAIC (Watanabe, 2010) is fully Bayesian in the sense that it is computed using the whole posterior distribution. It takes the computed log posterior predictive density in equation (23) and adjusts for overfitting by adding a correction for the effective number of parameters (Gelman et al., 2014).

$$WAIC = -2lppd + 2p_{\text{WAIC}}, \quad (21)$$

where

$$lppd = \log \prod_{i=1}^n p_{\theta|\mathbf{y}}(y_i) = \sum_{i=1}^n \log \int p(y_i|\theta) p(\theta|\mathbf{y}) d\theta \quad (22)$$

The left hand side term of equation (22), $p_{\theta|\mathbf{y}}(y_i)$ equals the integral on the right hand side because θ is unknown, and so we take an expectation of $p_{\theta|\mathbf{y}}(y_i)$ over θ given \mathbf{y} , i.e. we integrate out θ . In practice, this is computed as follows:

$$\text{computed } lppd = \sum_{i=1}^n \log \left(\frac{1}{S} \sum_{s=1}^S p(y_i | \theta^s) \right) \quad (23)$$

where θ^s , $s = 1, \dots, S$, are posterior samples from $p(\theta | \mathbf{y})$, and

$$p_{\text{WAIC}} = \sum_{i=1}^n \text{var}_{\theta|\mathbf{y}}(\log p(y_i | \theta)) \quad (24)$$

In practice, to compute the posterior variance for every term in the log predictive density $V_{s=1}^S \log p(y_i | \theta^s)$, where $V_{s=1}^S$ is the sample variance, we use: $V_{s=1}^S a_s = \frac{1}{S-1} \sum_{s=1}^S (a_s - \bar{a})^2$. The effective number of parameters is then given by the sum of $V_{s=1}^S$ over all the data points:

$$\text{computed } p_{\text{WAIC}} = \sum_{i=1}^n V_{s=1}^S (\log p(y_i | \theta^s)). \quad (25)$$

AIC, AICc, DIC, WAIC can be used for model selection through predictive accuracy, whereas BIC is used to approximate the marginal probability density of the data under the model $p(\mathbf{y})$, useful for discrete model selection when estimating the relative posterior probabilities (Gelman et al., 2014).

Out of AIC, AICc, DIC, WAIC, the latter one is preferred as it has better asymptotic properties. This is because in their derivation using the Taylor series expansion of the information distance between the model and the real system around the optimal parameter that minimises this distance, terms above the third order are neglected for WAIC, whereas for AIC for example, terms above second order are neglected, so the error associated with WAIC is lower, i.e. the predictive accuracy is higher for WAIC (Watanabe, 2015). In addition, WAIC is asymptotically equivalent to Bayesian leave-one-out cross-validation (LOO-CV) (Watanabe, 2010). In Bayesian Loo- CV, one point is removed at a time from the data, the model is trained on the $n - 1$ observations and the model fit to the data assessed by computing the log predictive density of the holdout data. A model that has a higher Bayesian LOO- CV estimate of the out-of-sample predictive fit (or lower -2 times this estimate, to be on the deviance scale) is preferred. The Bayesian LOO-CV estimate of the out-of-sample predictive fit is:

$$lppd_{\text{LOO-CV}} = \sum_{i=1}^n \log p(y_i | \mathbf{y}_{-i}) = \sum_{i=1}^n \log \int p(y_i | \theta) p(\theta | \mathbf{y}_{-i}) d\theta,$$

which in practice is calculated as:

$$\text{computed } lppd_{\text{LOO-CV}} = \sum_{i=1}^n \log \left(\frac{1}{S} \sum_{s=1}^S p(y_i | \theta^s) \right),$$

where θ^s is a sample from $p(\theta | \mathbf{y}_{-i})$.

AIC, AICc, DIC and WAIC are scores for the adjusted within-sample predictive accuracy, i.e. all points in \mathbf{y} are used to calculate the scores and prediction is done for the same points; the bias associated with doing this goes to zero as the number of observations increases, i.e. asymptotically. The scores use a penalty term for the model complexity. In contrast, LOO-CV measures the out-of-sample prediction accuracy, as its calculation relies on the points in \mathbf{y}_{-i} , i.e. on

all points, except the i^{th} , and we calculate the predictive accuracy for the i^{th} point. What is more, when comparing WAIC and DIC, WAIC can produce a reliable result even when dealing with multimodal posterior distributions, as it uses the entire posterior distribution, while DIC would be misleading as it uses the mean posterior estimate (e.g. in a bimodal posterior distribution, the mean would be between the two modes). It is known that DIC cannot deal with singular models (Watanabe, 2010).

5 | SIMULATIONS

The parameters that we wish to infer, described in detail in Section 2 are taken as arguments to the PDEs (1) - (2), and predicted flow and pressure are obtained along several locations in the 21 vessels. Since measured data are only available from one location in the main pulmonary artery, we only use the corresponding prediction from the MPA. We drive the system by the measured flow data, and compare our predictions to pressure measurements in the MPA. This is done using the residual sum-of-squares, S (equation (8)). The set of parameters that give the closest prediction to the measured data is our estimate (maximum likelihood estimate) (Paun et al., 2017). In practice, this is performed using the nonlinear constrained optimization scheme (SQP algorithm). The uncertainty around these estimates is obtained by approximately sampling from the posterior distribution using MCMC (DRAM, AM). We implement these algorithms by making use of the DRAM toolbox implemented in Matlab (Laine, 2007).

The parameters are on different scales: $f \in [10^4, 10^6]$, $r_1, r_2, c \in [-3, 2.5]$, $\xi \in [0, 0.5]$, and these parameter bounds are chosen to be biologically meaningful. To avoid having an ill-conditioned problem induced by a high condition number in the Hessian matrix (Yang and Lee, 2010), we rescale the parameters to have the same order of magnitude and use these scaled parameters in the optimization routine, as well as in the DRAM algorithm. There are certain parameter configurations outside the domain of convergence of the numerical scheme. This prompted us to assign a very high value to the RSS (10^{10}) that marks the unsuccessful simulations (data are unlikely given those parameter values).

The 20 different initial values for the parameters passed in the optimization algorithm are uniformly drawn from a Sobol sequence to ensure a good coverage of the multidimensional parameter space (Bratley and Fox, 1988). The SQP algorithm is iterated until it satisfies the convergence criterion, i.e. $|\theta_i - \theta_{i+1}| < 1e - 11$. Once convergence has been reached, we take the optimum parameter values as starting values for the DRAM algorithm in order to speed up the simulations.

Let us now define the likelihood, priors, proposal distribution and posterior distribution used in the DRAM algorithm.

- Data likelihood: $y_i | \theta \sim N(m_i(\theta), \sigma^2)$, i.e.

$$p(\mathbf{y} | \theta, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^n \exp \left(- \frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2\sigma^2} \right)$$

- Priors: $\theta \sim \text{Truncated } MVN(\mu, T)$, T : positive definite matrix, and due to lack of prior knowledge, we assign high variances to the parameters, to make the prior noninformative.

$\sigma^2 \sim \text{Inv-Gamma}(a, b)$, where following Algorithm 1, we denote $a = \frac{n_s}{2} > 0$, $b = \frac{n_s \gamma_s^2}{2} > 0$ to be the shape and scale parameters, with γ_s^2 being the prior for σ^2 and n_s the prior accuracy for γ_s^2 – a large value of n_s means we are certain of our prior, i.e. the prior dominates, and so the posterior samples will be close to σ_s^2 . The reason for choosing this prior is that the Inv-Gamma family is conditionally conjugate (i.e. if the prior distribution for σ^2 is an Inv-Gamma, then the conditional posterior distribution for σ^2 is also Inv-Gamma), allowing us to easily draw posterior samples for σ^2 from its full conditional distribution in a Gibbs step. We assume a noninformative prior for σ^2 , so we set

$n_s = 1$ and let γ_s^2 be the initial noise variance, σ_2 chosen by the user (the defaults in the toolbox).

$$p(\theta) = \det(2\pi T)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\theta - \mu)' T^{-1}(\theta - \mu)\right),$$

$$p(\sigma^2) = \frac{b^a}{\Gamma(a)} (\sigma^2)^{-a-1} \exp\left(-\frac{b}{\sigma^2}\right)$$

- Proposal distribution: $q(\theta^*, \theta^{k-1}) = \text{Truncated } MVN(\theta^{k-1}, V)$.

For θ^* inside boundaries:

$$q(\theta^*, \theta^{k-1}) = \det(2\pi V)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\theta^* - \theta^{k-1})' V^{-1}(\theta^* - \theta^{k-1})\right),$$

- Posterior distribution $\propto \text{Likelihood} \times \text{Prior}$, i.e.

$$p(\theta, \sigma^2 | \mathbf{y}) \propto p(\mathbf{y} | \theta, \sigma^2) p(\theta) p(\sigma^2)$$

which yields:

$$p(\theta, \sigma^2 | \mathbf{y}) \propto (\sigma^2)^{-\frac{n}{2}-a-1} \exp\left(-\frac{0.5 \sum_{i=1}^n (y_i - m_i(\theta))^2 + b}{\sigma^2}\right) \times$$

$$\exp\left(-0.5(\theta - \mu)' T^{-1}(\theta - \mu)\right).$$

In this expression, $\mathbf{m}(\theta)$ is the prediction from the PDEs and \mathbf{y} is the measured data.

DRAM is used to sample the parameters θ in a Metropolis-Hastings step. We then fix θ to the sample drawn, and we update σ^2 , the error variance, in a Gibbs step, by drawing samples from an Inverse-Gamma distribution, $\text{Inv-Gamma}\left(\frac{n}{2} + a, 0.5 \sum_{i=1}^n (y_i - m_i(\theta))^2 + b\right) = \text{Inv-Gamma}\left(\frac{n_s+n}{2}, \frac{n_s \gamma_s^2 + \sum_{i=1}^n (y_i - m_i(\theta))^2}{2}\right)$. We notice that when going from the prior to the posterior, the distribution is modified to include the contribution from the data, $\frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2}$, and the accuracy in the σ^2 sample increases when data become available (we add $\frac{n}{2}$ to the prior shape parameter - the more data points, the higher the accuracy). We use the θ samples as an input in the mathematical equations (1) - (2), and thus, for every DRAM sample we solve the PDEs. Since every forward simulation takes around 20s of CPU time, in the interest of time, DRAM is started from optimised values. In our analysis, we compare the pressure prediction obtained using the DRAM samples to the measured pressure.

To check for convergence of the Markov chains, we have employed the Geweke test, the Integrated Autocorrelation Time, the Effective Sample Size, and the Brooks Gelman Rubin test for sub-chains of the same chain started from optimised values. We acknowledge the fact that we do not satisfy a fundamental condition of this test (using parallel chains started from overdispersed values), but to make the method computationally viable, adopting these heuristics is indispensable.

Regarding model selection, we have compared the model with tapering (5D) to the model without the tapering factor (4D).

6 | RESULTS

This analysis is done on two mice, a control and a hypoxic mouse. For each mouse, we aim to compare two models: a 5D model which includes all the parameters considered (see (6)), and a 4D model (nested within the 5D model) which excludes the tapering factor ξ , as it is uncertain whether it is needed in the model.

6.1 | Optimization Results

We start our analysis with an exploration of the objective function S (equation (8)), calculated for pressure in the MPA. We can visualize the landscape of S in 1D and 2D space (in 2D we vary two parameters at the time, and in 1D we vary one parameter at the time, while keeping all the others fixed to some value). We choose a representative mouse (the control mouse) and illustrate some of the 2D plots of S in Figure 2 (4D model: no tapering) and Figure 3 (5D model: tapering).

S exhibits unimodality and it is highly skewed in 2D, and this is the case for both the control and the hypoxic mouse tested, for both models. In some of the figures, there is a sudden jump from low S values to high S values (10^{10}). These very high S value region marks the parameter domain that is invalid under the mathematical model. This shows that the biological parameter bounds are valid in an univariate sense, but this domain gets more restricted in higher dimensions, with some parameter combinations that produce no solutions to the PDEs (1) - (2), due to being outside the domain of convergence of the numerical scheme. The unimodality in S motivates applying the nonlinear constraint optimization to our problem to find the set of parameters that minimise S . We do so for all of the parameters in the model (see equation (6)), for both mice, and to ensure good coverage of the multidimensional space, we generate 20 initial values from a Sobol sequence. Our results reveal that regardless of the starting value, we manage to recover the same parameter values. control mouse: $f = 85529$; $r_1 = 1.96$; $r_2 = 0.25$; $c = -2.03$; $\xi = 0.26$, and $S_{\text{final}} = 104.5$; hypoxic mouse: $f = 228383$; $r_1 = 0.59$; $r_2 = 0.14$; $c = -0.22$; $\xi = 0.09$, and $S_{\text{final}} = 80$, and we consider these to be the maximum likelihood estimates. We can use these estimates as input into the PDEs to predict the pressure signal, and compare this against the measured pressure (Figure 4), and we do so for both mice.

Figure 4 shows the optimised pressure waveform for the 5D problem plotted alongside the measured pressure data for both mice. Panel (a) shows that in the case of the healthy mouse, the simulated pressure closely follows the measured pressure except near the peak, where an offset is registered. The overall model prediction appears to be better for the hypoxic mouse (panel (b)), which is also reinforced by the lower S score.

Next, we set $\xi = 0$ and repeat the analysis for a parameter set that excludes ξ . Our findings indicate that for the Control mouse, the fit is poorer if ξ is not included, but the fit is very similar for the Hypoxic mouse. Control mouse: $f = 100398$; $r_1 = 1.69$; $r_2 = 0.17$; $c = -1.30$, and $S_{\text{final}} = 126.5$; Hypoxic mouse: $f = 243645$; $r_1 = 0.52$; $r_2 = 0.13$; $c = -0.24$, and $S_{\text{final}} = 80.3$. See Figure 5 for a visual inspection of the fit.

However, we cannot decide on whether to include ξ in our model without a formal model selection analysis, which will follow later in the paper.

6.2 | MCMC Results

Next, we aim to quantify the uncertainty around our parameter estimates, and to do so, we implement the DRAM algorithm, the AM algorithm and the DR algorithm to our problem. This is done for each mouse for the two models (with and without the taper factor, ξ).

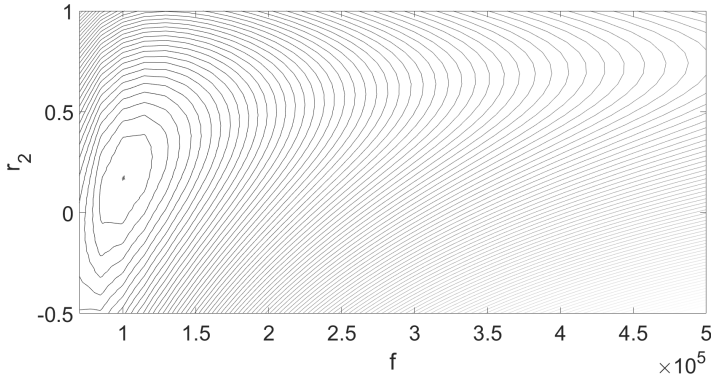
6.2.1 | AM Results

In the AM algorithm there are a couple of parameters that need to be set prior to the commencement of the algorithm. One such parameter is s_d (see (13)), which we take to be the recommended value in literature, $2.4^2/d$. Another such parameter is the length of the adaptation interval, t_{ad} (see (13)), which in practice is chosen to ensure mixing while having enough points that we use for updating the chain covariance matrix. The recommended t_{ad} is 100 in literature, however we have found that for our problem, a much larger t_{ad} was needed. We experimented with values of 100, 500 and 1000. Our findings indicate that $t_{ad} = 1000$ gives higher acceptance rates by up to 5% than $t_{ad} = 500$. The acceptance rate stays between 19% – 23%, and the mixing appears to be better than for $t_{ad} = 100$, so we have chosen $t_{ad} = 1000$. In Figures 6 to 10 (in the text), and 16 and 17 (in the Appendix) we show traceplots, marginal posterior densities and scatterplots of parameters sampled during the AM simulation for the two mice, for both models and Table 1 provides posterior summaries for the parameters, and we notice that the posterior means are very similar to the optimised values from the Optimization Results section.

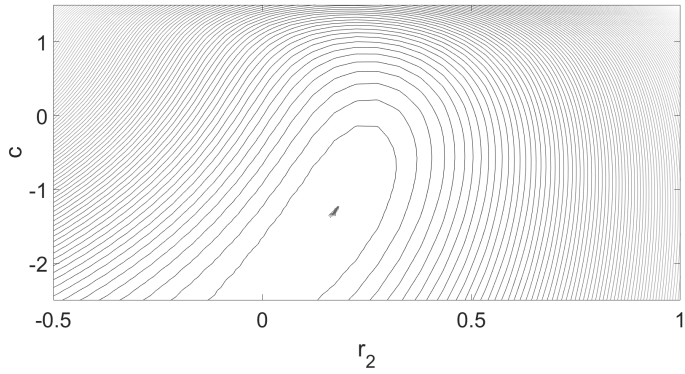
For the control mouse, for the two models (Figures 6 - 9 in the text and 16 in the Appendix), we can see that the chains fluctuate around the parameter values obtained from optimisation, suggesting that the chain has reached the highest posterior density region, and that the optimization might have found the global optimum.

There are no signs of multimodality in the objective function, which was reinforced by the convergence results we obtained through optimization. In addition, the S values are steady around the initial, lowest S value and the σ^2 chain does not indicate non-convergence (see Figure 7). Also, the marginal posterior densities for some parameters have long tails (Figure 8), which reinforces the skewness evident in the 2D S plots (Figures 2 and 3). The scatterplots (Figures 9) reveal strong correlations between some of the parameters, which is a reason why the algorithm needs longer runs to converge. For example, the stiffness f is highly correlated to the tapering factor, ξ , as the wall stiffness increases, the vessels will become less tapered (top and bottom vessel radii get closer together), which is what we would expect (see (5)). Another example is the high correlation between stiffness, f and compliance adjustment parameter, c , i.e. as the stiffness of the vessel wall increases, the compliance adjustment increases, and so the nominal compliance or elasticity decreases, which is expected (see (4)).

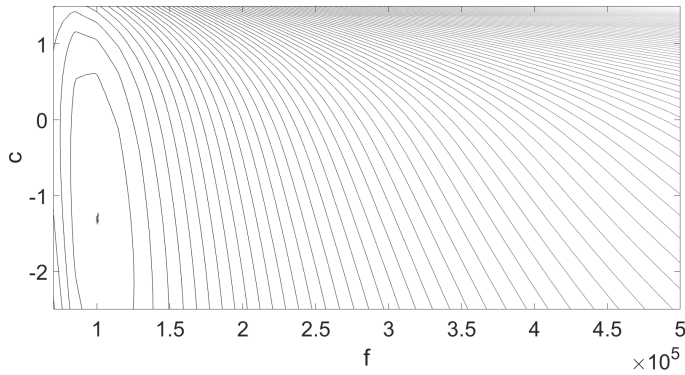
Regarding the hypoxic mouse, for the 5D model, we note a more pronounced skewness in the distribution for two of the parameters: f and ξ (Figure 10), which are strongly correlated. We also notice that the biological range for ξ is overly conservative, and a wider range should be used. In addition, the posterior mean appears to vary slightly from the maximum likelihood estimates for some of the parameters, but the two are more similar in the 4D model for the hypoxic mouse (Figure 17 in the Appendix).



(a) Residual sum-of-squares (S) for parameters f and r_2 with $S : (1.2 \times 10^2, 6.5 \times 10^4)$

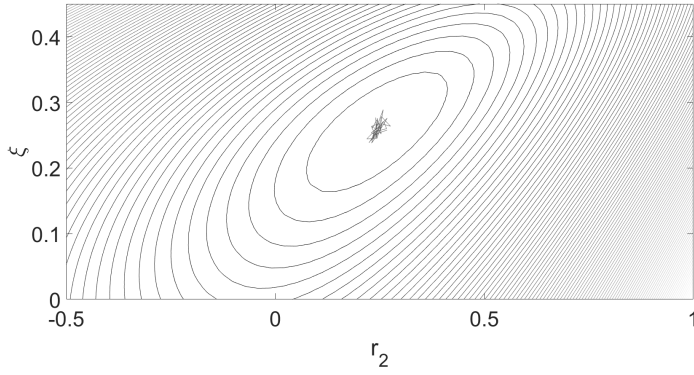


(b) Residual sum-of-squares (S) for parameters r_2 and c with $S : (1.2 \times 10^2, 2.2 \times 10^4)$

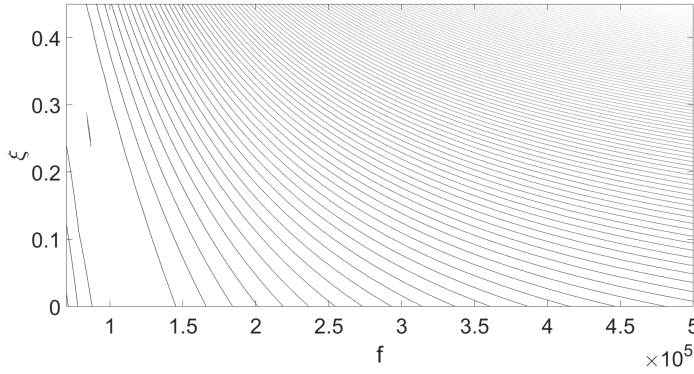


(c) Residual sum-of-squares (S) for parameters f and c with $S : (1.2 \times 10^2, 1.0 \times 10^5)$

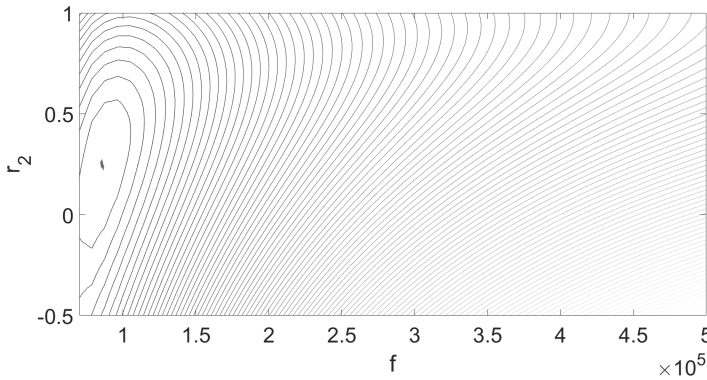
FIGURE 2 Contour plots of the objective function (residual sum-of-squares) in 2D, decreasing in the direction towards the inner slices for the 4D model corresponding to a control mouse. The black trajectories in the innermost slice indicate parameter (defined in the Simulations section) samples from the posterior distribution obtained using the Delayed Rejection Adaptive Metropolis algorithm. For graphical visibility, the innermost slice is not further resolved with additional contour lines.



(a) Residual sum-of-squares (S) for parameters r_2 and ξ with $S : (1.0 \times 10^2, 1.7 \times 10^4)$

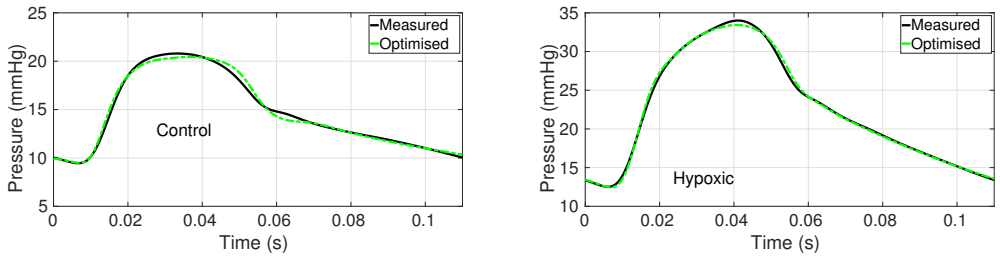


(b) Residual sum-of-squares (S) for parameters f and ξ with $S : (1.0 \times 10^2, 1.6 \times 10^5)$



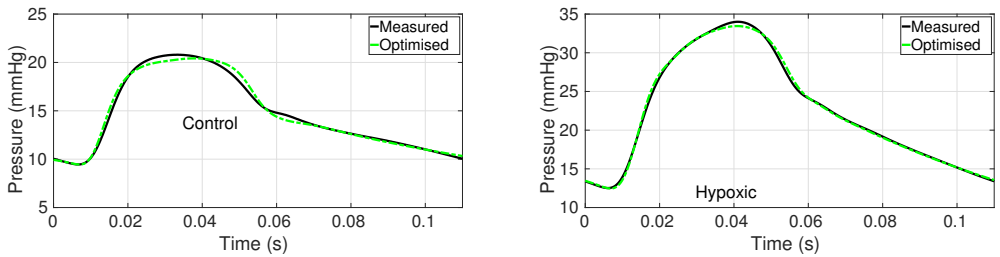
(c) Residual sum-of-squares (S) for parameters f and r_2 with $S : (1.1 \times 10^2, 1.1 \times 10^5)$

FIGURE 3 Contour plots of the objective function (residual sum-of-squares) in 2D, decreasing in the direction towards the inner slices for the 5D model corresponding to a control mouse. The black trajectories in the innermost slice indicate parameter (defined in the Simulations section) samples from the posterior distribution obtained using the Delayed Rejection Adaptive Metropolis algorithm. For graphical visibility, the innermost slice is not further resolved with additional contour lines.



(a) Pressure obtained from optimization (dashed line) plotted along-side the measured pressure (solid line) for the 5D model corresponding to a control mouse, $S = 104.5$
(b) Pressure obtained from optimization (dashed line) plotted along-side the measured pressure (solid line) for the 5D model corresponding to a hypoxic mouse, $S = 80$

FIGURE 4 Comparison between measured pressure (solid line) and pressure obtained from optimization (dashed line) for the 5D model for a control and a hypoxic mouse.



(a) Pressure obtained from optimization (dashed line) plotted along-side the measured pressure (solid line) for the 4D model corresponding to a control mouse, $S = 126.5$
(b) Pressure obtained from optimization (dashed line) plotted along-side the measured pressure (solid line) for the 4D model corresponding to a hypoxic mouse, $S = 80.3$

FIGURE 5 Comparison between measured pressure (solid line) and pressure obtained from optimization (dashed line) for the 4D model for a control and a hypoxic mouse.

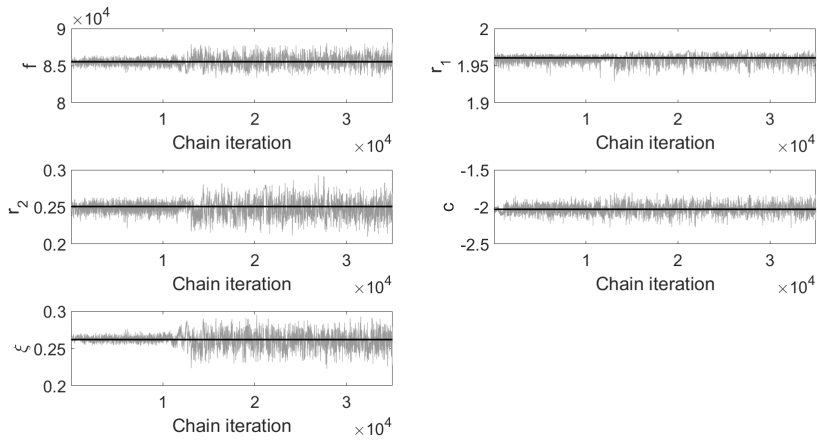


FIGURE 6 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 22%.

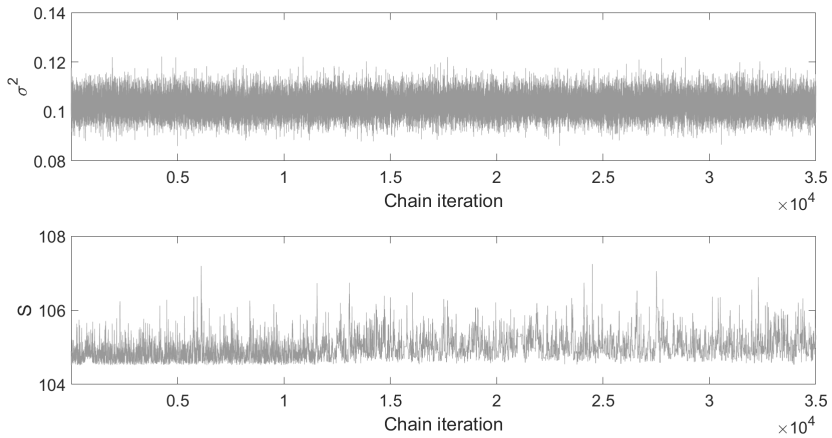


FIGURE 7 Markov chains traceplots for the error variance σ^2 , and the residual-sum-of-squares S obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse.

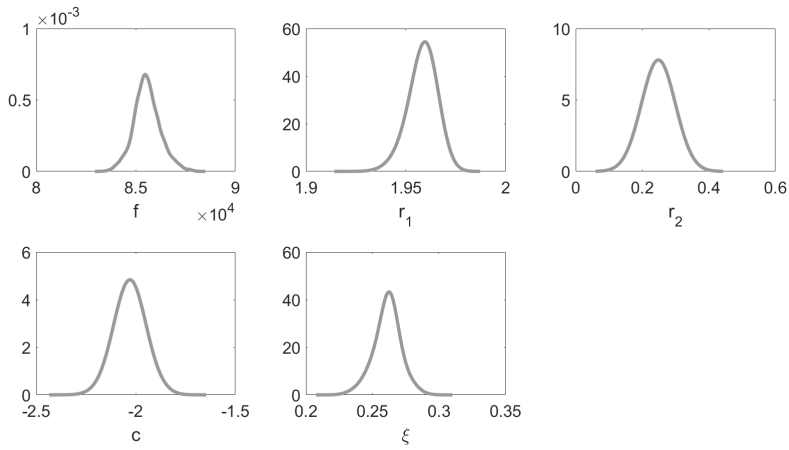


FIGURE 8 Marginal posterior densities of parameters (defined in the Simulations section) inferred using the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse. Plots were obtained using a kernel density estimator with customised bandwidth.

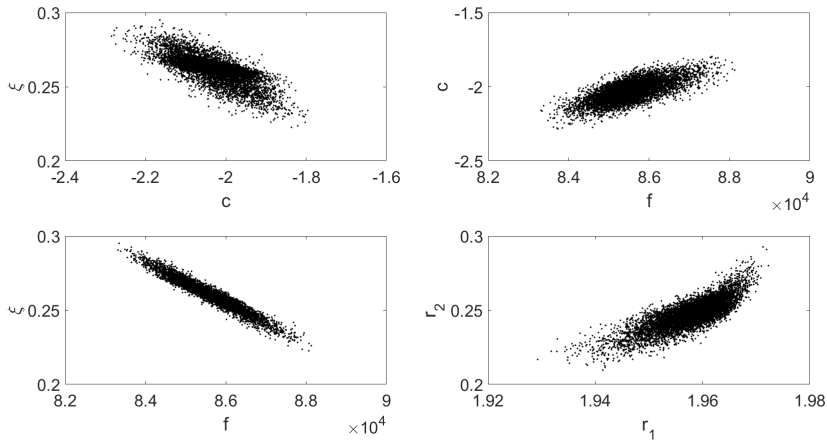


FIGURE 9 Scatterplots of parameter (defined in the Simulations section) chains obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse.

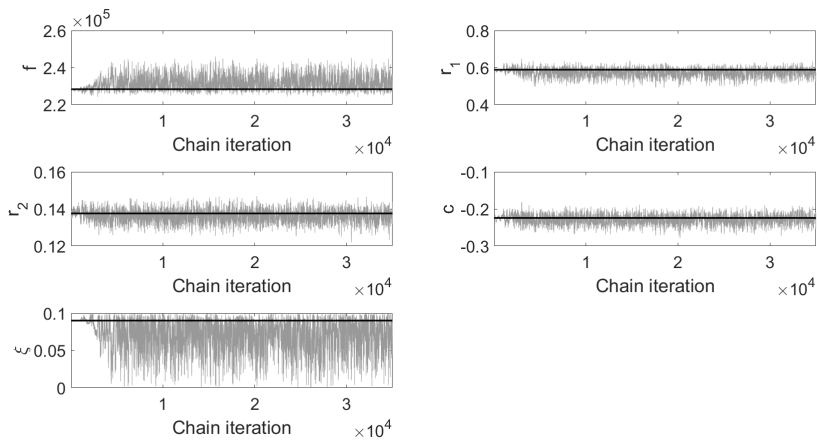


FIGURE 10 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a hypoxic mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 23%.

6.2.2 | DR Results

Figure 11 shows 10000 iterations of a chain produced using the DR algorithm for the control mouse, 5D model. Despite the acceptance rate which is close to the optimal target acceptance rate given in literature (23%), the plots reveal non-convergence of the chains, and the parameter space is explored slowly, leading to bad mixing. DR seems to be affected by the strong correlations between the parameters. Therefore, the chains would have to be run for a very long time, implying that the AM algorithm is essential in reaching convergence in a reasonable time frame.

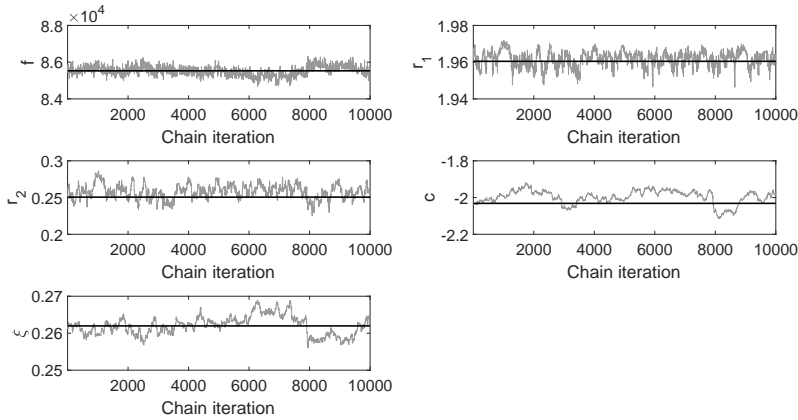


FIGURE 11 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Delayed Rejection algorithm for the 5D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 22%.

6.2.3 | DRAM Results

Lastly, we implement the DRAM algorithm starting from optimised values for every mouse and model type. We find that the estimates are very similar to the estimates obtained using the AM algorithm (see Table 1), which is expected. The algorithm used to sample draws is irrelevant if the samples are approximately drawn from the posterior distribution; the algorithm only influences the convergence of the chains. This is supported by the traceplots showing these samples (Figures 12 and 13 in the text and 18, 19, 20 in the Appendix), which do not indicate non-convergence, as the chains appear to be exploring the space around the optimum values well and are steady around these values. We find that the DRAM algorithm has increased the acceptance rate compared to the AM algorithm (from 20% up to 40%). This is not surprising, considering the extra DR step in the DRAM algorithm, which proposes a second point at every iteration of the algorithm. This clearly increases the acceptance rate, and thus, fewer iterations are needed to reach convergence when compared to AM. This, however, does not imply less computational time, since in every DRAM iteration, when a second point is proposed, the PDEs have to be solved twice. Therefore, AM and DRAM are computationally equivalent in our case, since double the number of iterations were needed for AM to decrease the correlation between chain samples. The inspection of the convergence diagnostics (discussed in the MCMC Convergence Diagnostics section below), as well as the improvement in the acceptance rate has made us favour DRAM over AM.

Let us now illustrate (Figures 14 and 15) what happens if we use the DRAM algorithm in (Smith, 2013), that does

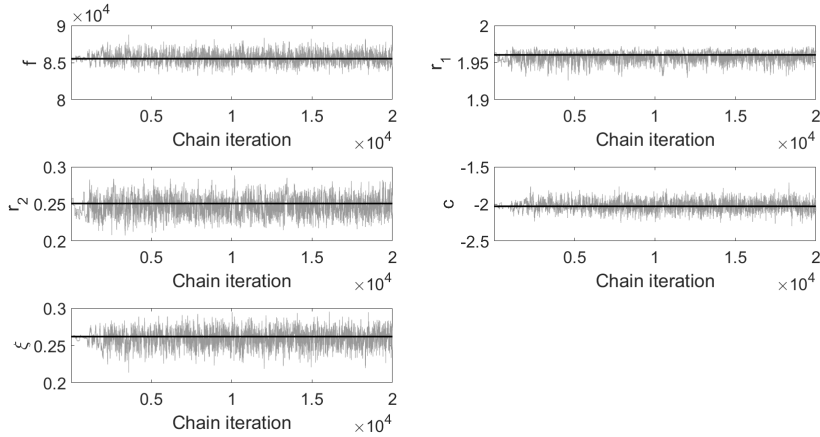


FIGURE 12 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Delayed Rejection Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 39%.

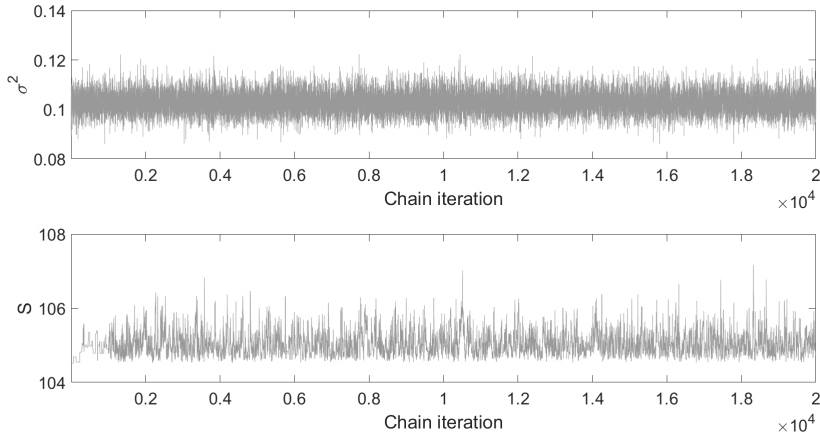


FIGURE 13 Markov chains traceplots for the error variance σ^2 , and the residual-sum-of-squares S obtained using the Delayed Rejection Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse.

not work in the transformed parameter space when calculating the acceptance probabilities, in particular the 2nd stage acceptance probability.

When looking at Figure 15, it is readily apparent that several proposals with high S are being falsely accepted. Figure 13 for the correct DRAM algorithm does not display such spikes, and the traceplots in Figure 12 are well-behaved. We falsely accept points because the ratio of the proposal distributions in equation (12) becomes very large, since it is calculated in the unscaled parameter space, and the starting values (optimum values) are very close to the boundaries or to the invalid parameter domain (see Figure 3). This implies that many of the proposed points will either lie outside

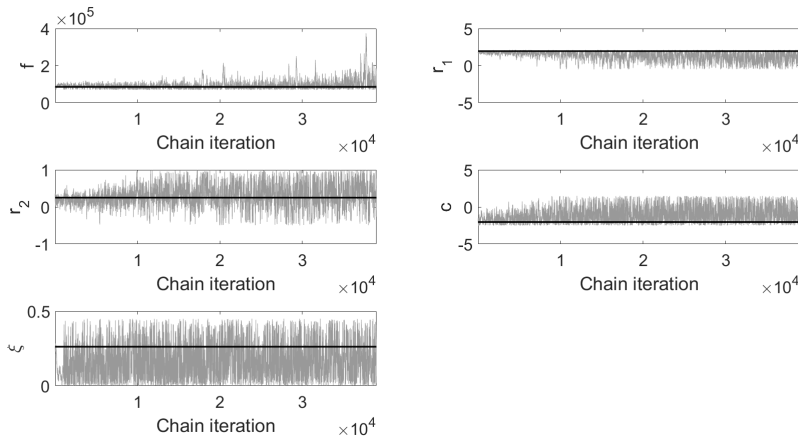


FIGURE 14 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the original Delayed Rejection Adaptive Metropolis algorithm in Smith (2013) without the correction term discussed in this paper, for the 5D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 30%.

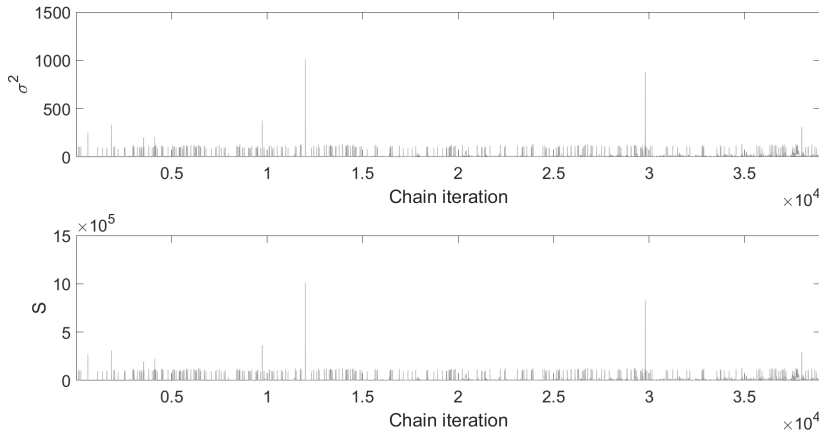


FIGURE 15 Markov chains traceplots for the error variance σ^2 , and the residual-sum-of-squares S obtained using the original Delayed Rejection Adaptive Metropolis algorithm in Smith (2013) without the correction term discussed in this paper, for the 5D model corresponding to a control mouse.

boundaries or inside the invalid domain, making f , the parameter with dominating magnitude, and S become high. When this ratio becomes large, the proposed point will be falsely accepted, as the acceptance rate is the minimum between 1 and the log of a high number, i.e. the acceptance probability will be 1. This would however not happen if the optimum value was somewhere far from the boundaries or the invalid domain, because the algorithm would explore a region of fairly similar values. This is tested for the case of the control mouse, model 4D, and the findings (not shown here) back

up our hypothesis.

6.3 | MCMC Convergence Diagnostics

It is now of interest to perform convergence diagnostics on the DRAM chains for every mouse, for the two models. To do so, we employ the Geweke test, the Brooks Gelman Rubin test, the Integrated Autocorrelated Time (IACT), and the Effective Sample Size (ESS) tests and the results are summarised in Table 2.

The Geweke test reveals a high p-value (0.99), well above 0.05 for all 4 scenarios, suggesting that there is no evidence of a difference between the mean of the first 10% samples and last 50% samples in the chain. The MPSRF is below 1.1 for all 4 cases, the highest one being 1.03 and the lowest 1.0004. The 4D model for both mice provides a lower MPSRF (< 1.01) than the 5D model. Nevertheless, the within and between chain covariance matrices are similar for both models. The estimates for IACT and ESS suggest that there is some correlation still left between the samples, in particular for the 5D model, where we have seen that there is a very strong correlation between f and ξ . This strong correlation delays the algorithm from sampling independent draws. For the 4D model, for both mice, we notice that on average, one independent sample is drawn every 13 samples. Therefore, the ESS is somewhere in the range [1200, 1800] out of a total number of 20000 iterations. For the 5D model, there is a more pronounced correlation effect, with an average IACT of 27 and ESS ranging between [700, 1500]. While the Geweke test and the MPSRF do not indicate non-convergence of the chains, the IACT values (slightly too high) and ESS values (slightly too low) indicate that the DRAM algorithm should be run for longer in order to decrease the correlation between samples.

When analysing Table 3 which displays the MCMC convergence diagnostics for the AM algorithm, we notice that the AM algorithm produces even more correlated samples, the IACT being consistently higher than that for the DRAM algorithm.

6.4 | Model Selection

We further consider model selection, since we have two competing models for the two mice: a 4D model (without tapering factor, ξ in), and a 5D model (with ξ in). For the purpose of model selection, we show the performance of 4 criteria: AICc, BIC, DIC and WAIC. We have seen in the previous section that there is still some correlation left between the chain samples. We therefore only consider a subset of 1000 samples of the total number of samples when computing WAIC. Table 4 summarises our model selection scores.

The results indicate that the scores are similar and fairly consistent across different criteria. For the control mouse, all the scores are lower for the 5D model, compared to the 4D model, which implies that in the case of the control mouse, adding the tapering factor, ξ makes the model more appropriate for the data, and so, it is worth including ξ in the model. On the other hand, for the hypoxic mouse, the scores are very similar across the two models, WAIC favours the 4D model, while AICc, BIC and DIC favour the 5D model. Considering the very low difference in S between the 4D and 5D models (0.56), and the fact that there is low difference between the scores, then we choose the 4D model to be used in the case of the hypoxic mouse. Hence, there is evidence that the tapering factor is not significant for the hypoxic mouse, but the opposite is true for the control mouse. A possible explanation for this could be that in hypoxia, the vessel walls become stiff, and so there will not be a pronounced decrease in the radii along the vessels' length.

7 | DISCUSSION

In this paper we have applied the Delayed Rejection Adaptive Metropolis algorithm and its variations (Adaptive Metropolis and Delayed Rejection) to measure uncertainty of the parameter estimates in a partial differential equations system of the pulmonary circulation. The method, which builds on and further improves existing literature by allowing for novel parameter scaling due to different magnitudes of the parameters has been applied on real data, coming from a healthy and a diseased mouse. Making a few restrictive assumptions about the parameters has allowed us to decrease the parameter dimension considerably (from a 55D problem – $21f, 11r_1, 11r_2, 11c, 1\xi$ parameters, down to a 5D problem – $1f, 1r_1, 1r_2, 1c, 1\xi$). Further studies should be done to check these assumptions. Some preliminary exploration of non-constant Windkessel adjustments, r_1 , r_2 , c and stiffness f on a smaller network has indicated that fitting different sets of parameters for every vessel would result in data overfitting. Even if the goodness of fit has improved, the model complexity has increased significantly. Model selection criteria have revealed that a very low error variance would be needed to favour use of the more complex model. Since our error variance is believed to be higher, we have rejected the complex model. The parameters had different orders of magnitude, which motivated us to improve existing DRAM code from the literature in order to assess uncertainty in a system of partial differential equations, which is the pulmonary circulation. To make the computations feasible, we have started the MCMC algorithm from optimised parameter values (a nonlinear constraint optimisation scheme was used). Our findings indicate that DRAM is preferred over DR and AM. The DR algorithm alone is very slow in converging, and the chains appear far from converging even after 10000 iterations. The AM algorithm retains an acceptance rate of about 20%, which appears to be too low, and there is correlation between the samples even after 36000 iterations. The DRAM algorithm appears to be the best, with an acceptance rate of about 40%, and fairly good mixing. While the Geweke test and the Multivariate Scale Reduction Factor test did not indicate non-convergence, the Integrated Autocorrelation Time appeared to be slightly too high after 20000 iterations. The strong dependence between some of the parameters makes the algorithm slow in generating more independent samples, and should therefore be run for longer. Another focus of this paper was to perform model selection to identify whether one of the parameters should be included in the model. The AICc, BIC, DIC and WAIC criterion consistently chose the more complex model for the control mouse. For the hypoxic mouse, the scores were inconsistent, but since the difference between them was low, and the most reliable score, WAIC, favoured the simpler model, we preferred this model. All these results are based on a mathematical model that looks more appropriate for the hypoxic mouse than for the control mouse. Possible causes for the slight model mismatch are: (i) the simplicity of the model specifying the elastic behaviour of the blood vessels and/or the boundary conditions, (ii) uncertainty of the geometry measurements which are not specific to a given mouse, (iii) a combination of (i) and (ii). This slight model mis-specification has the undesired effect of breaking the i.i.d. error Normality assumption. Therefore, to account for the correlated structure in the errors, we aim to use a Markov model that will also model the correlation structure, and not only the parameters and the error variance. In addition, future work will include implementing faster methods to infer the relevant parameters. This includes the Hamiltonian Monte Carlo algorithm (Neal, 2012), which suppresses the random behaviour of the algorithm by using a momentum variable that leads the direction in which the proposals are made. To speed up the simulations, we will run this algorithm on an objective function emulated using Gaussian Processes (Rasmussen and Williams, 2006). Another alternative to learn the unobserved parameters is by employing the Ensemble Kalman filter, following the idea in Arnold et al. (2017).

ACKNOWLEDGEMENTS

This work is part of the research programme of the Centre for multiscale soft tissue mechanics with application to heart & cancer (SoftTMech), funded by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, grant reference number EP/N014642/1. Olufsen, Haider and Qureshi were supported by National Science Foundation (NSF-DMS # 1615820). Data were made available by N. Chesler, Department of Biomedical Engineering, University of Wisconsin, Madison.

REFERENCES

- Akaike, H. (1971) Information theory and an extension of the maximum likelihood principle. In *Proc. 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR*, 267–281. Budapest: Akadémiai Kiadó.
- Arnold, A., Battista, C., Bia, D., Z. G. Y., Armentano, R. and Tran, H.T., O. M. (2017) EnKF based inflow estimator for a 1d arterial network. *J Verification, Validation and UQ*, **2**.
- Boggs, P. and Tolle, J. (2000) Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, **124**, 123 – 137. URL: <http://www.sciencedirect.com/science/article/pii/S0377042700004295>. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.
- Bratley, P. and Fox, B. (1988) Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.*, **14**, 88–100. URL: <http://doi.acm.org/10.1145/42288.214372>.
- Brooks, S. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434–455. URL: <http://www.tandfonline.com/doi/abs/10.1080/10618600.1998.10474787>.
- Burnham, K. and Anderson, D. (2002) *Model Selection and Multimodel Inference: A practical information-theoretic approach*. Springer-Verlag, second edn.
- Christen, J. and Fox, C. (2010) A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis*, **5**, 263–282.
- Eck, V., G., Donders, W., P., Sturdy, J., Feinberg, J., Delhaas, T., Hellevik, L., R. and Hub, W. (2016) A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications. *International Journal for Numerical Methods in Biomedical Engineering*, **32**, e02755–n/a. URL: <http://dx.doi.org/10.1002/cnm.2755>. E02755 cm.2755.
- Gelman, A., Hwang, J. and Vehtari, A. (2014) Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, **24**, 997–1016. URL: <http://dx.doi.org/10.1007/s11222-013-9416-2>.
- Gelman, A. and Rubin, D. B. (1992) Inference from iterative simulation using multiple sequences. *Statist. Sci.*, **7**, 457–472. URL: <http://dx.doi.org/10.1214/ss/1177011136>.
- Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *IN BAYESIAN STATISTICS*, 169–193. University Press.
- Gilks, W., Richardson, S. and Spiegelhalter, D. (1995) *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. Taylor & Francis. URL: http://books.google.com/books?id=TRXrMWY_i2IC.
- Green, P. and Mira, A. (2001) Delayed rejection in reversible jump Metropolis–Hastings. *Biometrika*, **88**, 1035–1053. URL: <http://www.jstor.org/stable/2673700>.
- Haario, H., Laine, M., Mira, A. and Saksman, E. (2006) DRAM: Efficient adaptive MCMC. *Statistics and Computing*, **16**, 339–354. URL: <http://dx.doi.org/10.1007/s11222-006-9438-0>.
- Haario, H., Saksman, E. and Tamminen, J. (2001) An adaptive Metropolis algorithm. *Bernoulli*, **7**, 223–242.
- Kass, R., B.P., C., Gelman, A. and Neal, R. (1998) Markov Chain Monte Carlo in Practice: A Roundtable Discussion. *The American Statistician*, **52**, 93–100. URL: <http://amstat.tandfonline.com/doi/abs/10.1080/00031305.1998.10480547>.
- Krenz, G. and Dawson, C. (2003) Flow and pressure distributions in vascular networks consisting of distensible vessels. *Am J Physiol*, **284**, H2192–H2203.
- Laine, M. (2007) MCMC Toolbox for Matlab. <http://helios.fmi.fi/ainema/dram/>.

- Lax, P. and Wendroff, B. (1960) Systems of conservation laws. *Communications on Pure and Applied Mathematics*, **13**, 217–237. URL: <http://dx.doi.org/10.1002/cpa.3160130205>.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953) Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**, 1087–1092.
- Mira, A. (2001a) On Metropolis–Hastings algorithm with Delayed Rejection Metron. **LIX**, 231–241.
- Neal, R. M. (2012) MCMC using Hamiltonian dynamics. *ArXiv e-prints*.
- Noordegraaf, A. and Gali, N. (2011) The role of right ventricle in pulmonary arterial hypertension. *Eur Respir Rev*, **20**, 243–253.
- Paun, L., M., Qureshi, M., U., Colebank, M., Haider, M., Olufsen, M., Hill, N. and Husmeier, D. (2017) Parameter inference in the pulmonary circulation of mice. In *Proceedings of the 32nd IWSM*, vol. 1, 190–195.
- Qureshi, M., U., Colebank, L., M., Paun, M., Chesler, N., Haider, M., A., Hill, N., A., Husmeier, D. and Olufsen, M., S. (2017a) A computational study of pulmonary hemodynamics in healthy and hypoxic mice. URL: <https://arxiv.org/abs/1712.01699>.
- Qureshi, M., U., Haider, M., Chesler, N. and Olufsen, M. (2017b) Simulating effects of hypoxia on pulmonary haemodynamics in mice. In *Proc. 5th International Conference on Computational & Mathematical Biomedical Engineering (CMBE 2017)*, vol. 1, 271–274. Zeta Computational Resources Ltd.
- Rasmussen, C. and Williams, C. (2006) *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press.
- Schwarz, G. (1978) Estimating the dimension of a model. *Ann. Statist.*, **6**, 461–464. URL: <http://dx.doi.org/10.1214/aos/1176344136>.
- Shi, Y., Lawford, P. and Hose, R. (2011) Review of zero-d and 1-d models of blood flow in the cardiovascular system. *Biomed Eng Online*, **10**.
- Smith, R. (2013) *Uncertainty Quantification: Theory, Implementation, and Applications*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.
- Spiegelhalter, D., Best, N., Carlin, B. and Van Der Linde, A. (2002) Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**, 583–639. URL: <http://dx.doi.org/10.1111/1467-9868.00353>.
- Tabima, D., Roldan-Alzate, A., Wang, Z., Hacker, T., Molther, R. and Chesler, N. (2012) Persistent vascular collagen accumulation alters hemodynamic recovery from chronic hypoxia. *J. Biomech*, **45**, 799–804.
- Tierney, L. and Mira, A. (1999) Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, **18**, 2507–2515. URL: [http://dx.doi.org/10.1002/\(SICI\)1097-0258\(19990915/30\)18:17/18<2507::AID-SIM272>3.0.CO;2-J](http://dx.doi.org/10.1002/(SICI)1097-0258(19990915/30)18:17/18<2507::AID-SIM272>3.0.CO;2-J).
- Wang, Z. and Chesler, N. (2011) Pulmonary vascular wall stiffness: An important contributor to the increased right ventricular afterload with pulmonary hypertension. *Pulm. Circ*, **1**, 212–223.
- Watanabe, S. (2010) Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *CoRR*, **abs/1004.2316**. URL: <http://arxiv.org/abs/1004.2316>.
- (2015) Bayesian cross validation and WAIC for predictive prior design in regular asymptotic theory. *CoRR*, **abs/1503.07970**. URL: <http://arxiv.org/abs/1503.07970>.
- Wilson, R. (1967) *Simplicial Method for Convex Programming*. Ph.D. thesis, Harvard University.
- Yang, K. and Lee, T. (2010) Heuristic scaling method for efficient parameter estimation. *Chemical Engineering Research and Design*, **88**, 520 – 528. URL: <http://www.sciencedirect.com/science/article/pii/S0263876209002731>.

| Type | f | r_1 | r_2 | c | ξ | σ^2 | S |
|---------------|------------------|-----------------|-----------------|-----------------|----------------|-----------------|-------|
| Control 4D | 100374 (391) | 1.69 (0.01) | 0.17 (0.005) | -1.30 (0.04) | - | 0.12 (0.005) | 127 |
| Control 5D | 85471 (658) | 1.96 (0.006) | 0.25 (0.01) | -2.03 (0.07) | 0.26 (0.01) | 0.10 (0.004) | 105 |
| Hypoxic 4D | 243653 (1180) | 0.52 (0.01) | 0.13 (0.003) | -0.24 (0.01) | - | 0.08 (0.003) | 80.56 |
| Hypoxic 5D | 231627 (3720) | 0.57 (0.02) | 0.14 (0.003) | -0.23 (0.01) | 0.07 (0.02) | 0.08 (0.003) | 80 |

TABLE 1 Parameter (f, r_1, r_2, c, ξ) estimates and σ^2 estimate with their standard deviations inside brackets and S (residual-sum-of squares) for every type of mouse (control and hypoxic) and model (4D and 5D) obtained from running the Adaptive Metropolis algorithm. These parameters are defined in the Simulations section.

| Type | Geweke test (p-values) | MPSRF | IACT | ESS (out of 20000) |
|------------|------------------------|-------|------------------|-------------------------|
| Control 4D | all > 0.05 | 1.004 | (11,12,11,14) | (1443,1362,1424,1180) |
| Control 5D | all > 0.05 | 1.01 | (22,31,28,21,22) | (912,639,702,930,927) |
| Hypoxic 4D | all > 0.05 | 1.006 | (14,15,14,11) | (1412,1333,1465,1807) |
| Hypoxic 5D | all > 0.05 | 1.03 | (23,22,13,17,27) | (862,897,1530,1144,732) |

TABLE 2 MCMC Convergence Diagnostics: Geweke test, Multivariate Potential Scale Reduction Factor (MPSRF), Integrated Autocorrelation Time (IACT) for each of the parameters, Effective Sample Size (ESS) for each of the parameters, corresponding to the results obtained using the Delayed Rejection Adaptive Metropolis algorithm for the two mice (control, hypoxic) and the two models (4D containing 4 parameters and 5D containing 5 parameters).

| Type | Geweke test (p-values) | MPSRF | IACT | ESS (out of 36000) |
|------------|------------------------|-------|------------------|-------------------------|
| Control 4D | all > 0.05 | 1.01 | (22,22,20,22) | (1630,1671,1811,1625) |
| Control 5D | all > 0.05 | 1.04 | (39,47,49,27,43) | (918,767,731,1317,837) |
| Hypoxic 4D | all > 0.05 | 1.004 | (17,18,18,18) | (2123,2030,2056,1990) |
| Hypoxic 5D | all > 0.05 | 1.06 | (49,37,19,26,58) | (730,965,1899,1375,622) |

TABLE 3 MCMC Convergence Diagnostics: Geweke test, Multivariate Potential Scale Reduction Factor (MPSRF), Integrated Autocorrelation Time (IACT) for each of the parameters, Effective Sample Size (ESS) for each of the parameters, corresponding to the results obtained using the Adaptive Metropolis algorithm for the two mice (control, hypoxic) and the two models (4D containing 4 parameters and 5D containing 5 parameters).

| Type | AICc (AIC) | BIC | DIC | WAIC |
|------------|------------|-----|-----|------|
| Control 4D | 772 | 792 | 772 | 774 |
| Control 5D | 579 | 603 | 578 | 580 |
| Hypoxic 4D | 307 | 326 | 306 | 304 |
| Hypoxic 5D | 301 | 325 | 300 | 310 |

TABLE 4 Model selection scores (AICc, BIC, DIC, WAIC) for the 4D and 5D models for the control and the hypoxic mouse. AICc is very similar to AIC (up to 4 dp), since the sample size is large (1024 data points). For the calculation of AICc and BIC, we have estimated $\hat{\sigma}^2 = \frac{S}{n-d}$, for DIC, $\hat{\sigma}^2$ is the posterior mean of the σ^2 samples, and for WAIC, every σ^2 sample in the chain is used.

Appendices

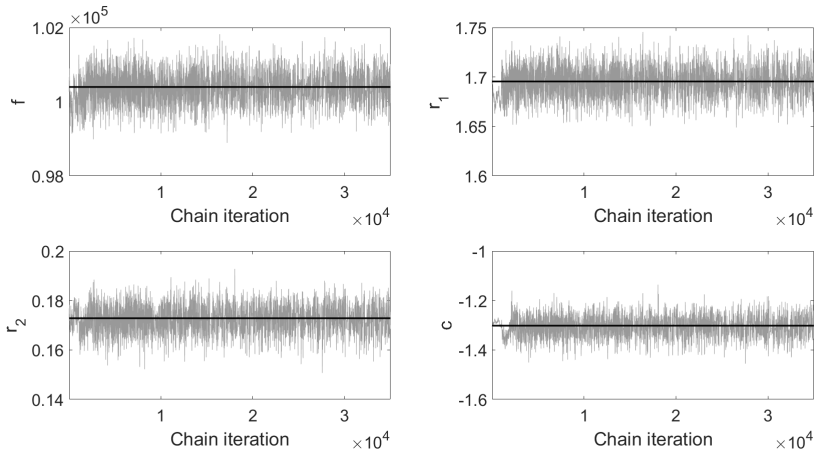


FIGURE 16 Markov chains traceplots for the parameters obtained using the Adaptive Metropolis algorithm for the 4D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 20%.

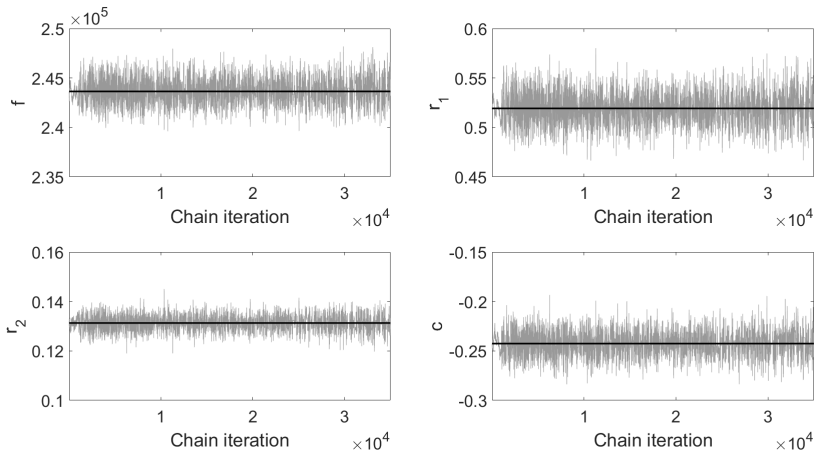


FIGURE 17 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Adaptive Metropolis algorithm for the 4D model corresponding to a hypoxic mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 19%.

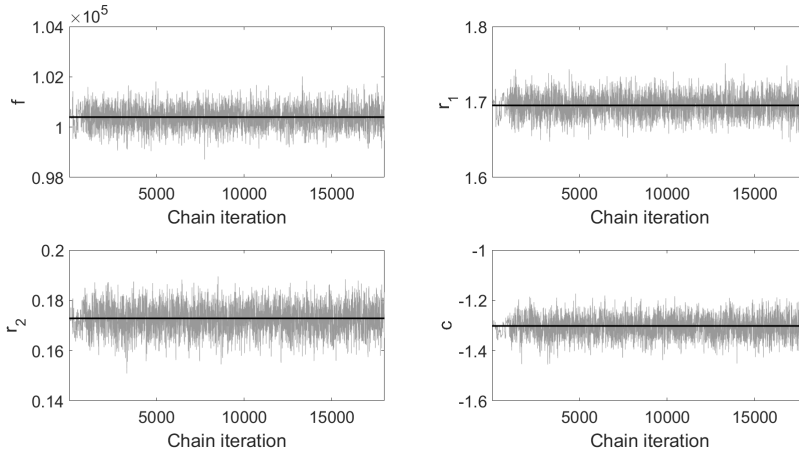


FIGURE 18 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Delayed Rejection Adaptive Metropolis algorithm for the 4D model corresponding to a control mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 48%.

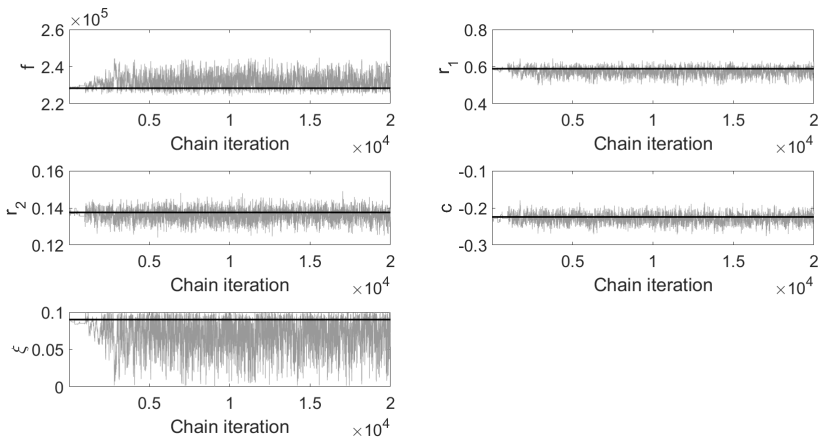


FIGURE 19 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Delayed Rejection Adaptive Metropolis algorithm for the 5D model corresponding to a hypoxic mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 45%.

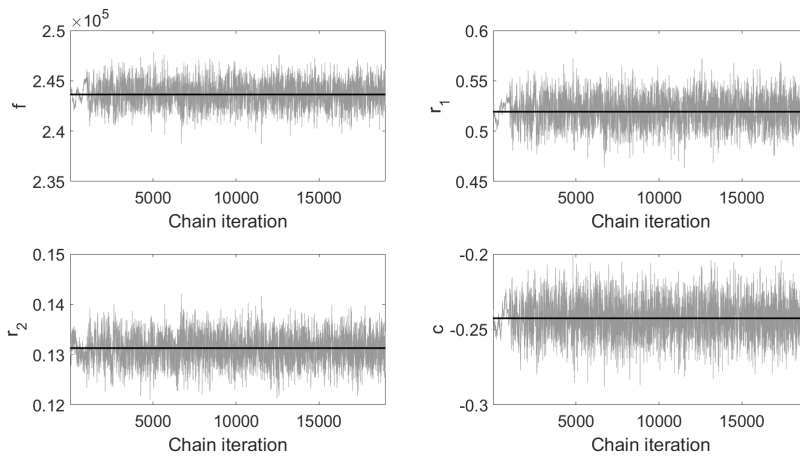


FIGURE 20 Markov chains traceplots for the parameters (defined in the Simulations section) obtained using the Delayed Rejection Adaptive Metropolis algorithm for the 4D model corresponding to a hypoxic mouse. Starting values for the algorithm are the optimised values and are superimposed in black horizontal lines. Acceptance rate is 46%.